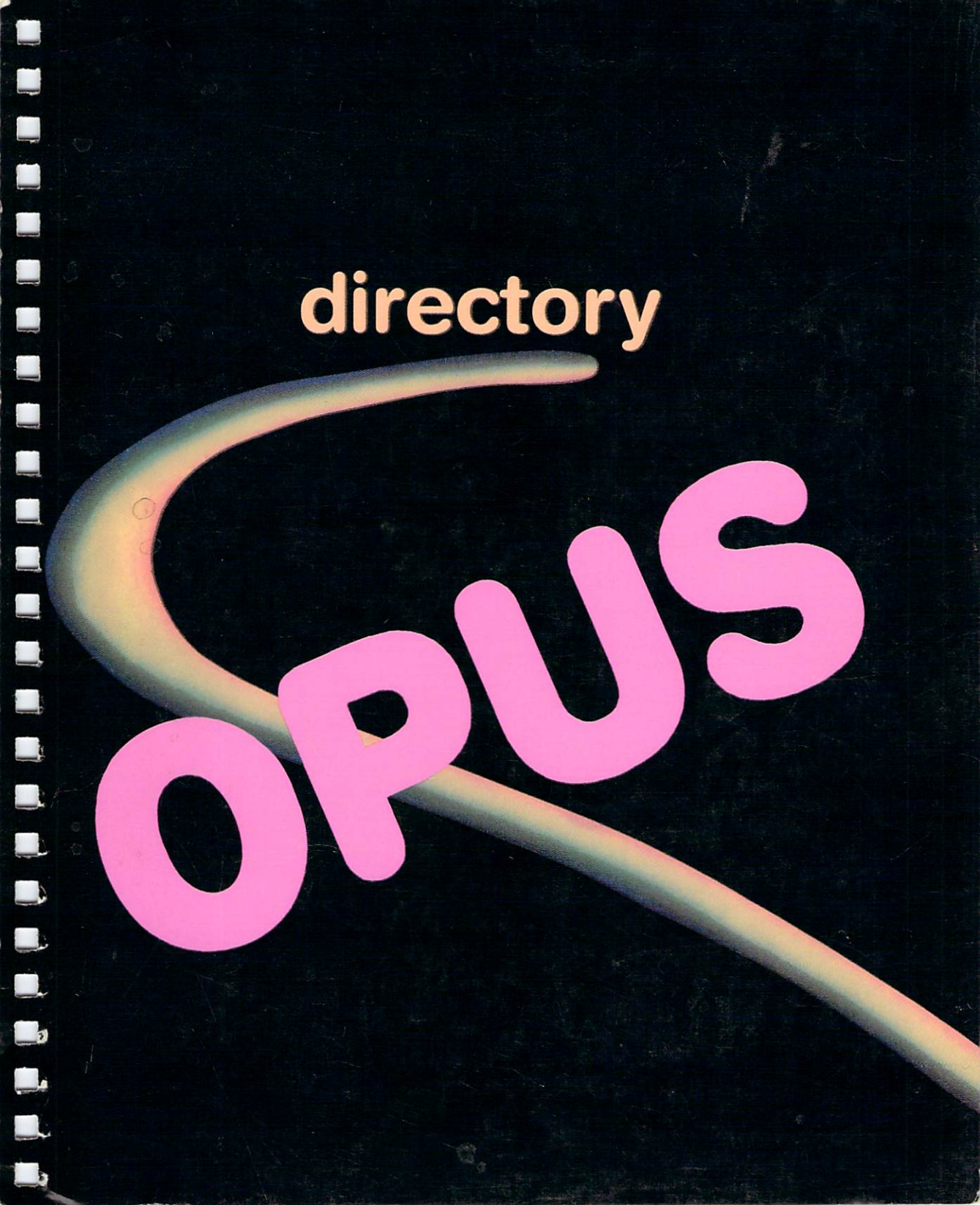

directory
OPUS

Directory

OPUS

By Jonathan Potter

from

INOVAtronics

Opus Legalities

All software programs and documentation provided herein are entirely copyright 1991 by **Jonathan Potter**, with the following exceptions:

1. cando.library is copyright 1991 by INOVAtronics, Inc.

2. stopus.library is copyright 1991 by INOVAtronics, Inc.
   stopus.library was written by Peter S. Martson.

3. arp.library by "Charlie Heath, Scott Ballantyne and the rest of the ARP guys."

4. req.library by Colin Fox (Pyramyd Designs) and Bruce Dawson (CygnusSoft Software).
   Copyright 1989, freely distributable.

All rights reserved. The programs and documentation are sold "AS IS" and without warranties as to performance, merchantability, or fitness for a particular purpose. Sale of this software conveys a license for its use on up to two computers owned or operated only by the purchaser. Copying this software or documentation by any means whatsoever for any other purpose is strictly prohibited.

Trademarks: "**Opus**", "**Directory Opus**", "**INOVAtronics**", "**CanDo**", and the "directory OPUS" logo are trademarks and service marks of **Inovatronics, Inc.**

"**Amiga**", "**AmigaDOS**", and "**Workbench**" are registered trademarks of **Commodore-Amiga, Inc.**

"**ARexx**" is a trademark of William S. Hawes

For Technical Support, Call the Opus Help Line: 1-214-340-4992

Directory Opus - Table of Contents

One - Introduction **1-1**

Files and Directories 1-1
Theory 1-4
Installing Opus 1-4
Running Opus 1-5
Common Terms 1-6

Two - Preliminary Information **2-1**

Aborting 2-1
Selecting Active Window 2-1
Reading a Directory 2-2
Entering Sub-Directories 2-3
Moving Through Buffers 2-3
Moving Throgh Dir List 2-3
Selecting Entries 2-5

Three - Default Gadgets **3-1**

Default Gadget Functions 3-1
The Tiny Gadgets 3-13

Four - Default Menus **4-1**

Five - Functions **5-1**

Six - Configuration **6-1**

Operation 6-1
Gadgets 6-6
Menus 6-11
Drives 6-12
File Types 6-13
Functions 6-17
Screen 6-19
System 6-23
Store Config 6-26
Quick Save 6-27

Seven - Launching External Programs **7-1**

Eight - On-line Help **8-1**

Nine - General Information **9-1**

Ten - ARexx Interface **10-1**

Function (See Functions, Chapter Six) ARexx equivalents 10-1
Independent ARexx Commands 10-8

Appendix A - The Conversion Program **A-1**

Appendix B - Key Codes and Qualifiers **B-1**

Index - Main **MainIndex-1**

Index - ARexx **ARexxIndex-1**

Chapter 1:
Introduction

A "directory utility" is, in its simplest definition, a program that allows you to manipulate files and directories.

If the terms files and directories are foreign to you, you should read the following section entitled Files and Directories - a Brief Introduction. You should probably also buy a good book on the workings of AmigaDOS, and possibly attend a local user group.

The first directory utility available for the Amiga was, I believe, a program called DirUtil, by Chris Nicotra (if this is not the case, my apologies to the original author). The directory utilities of old were simple affairs, consisting of a single window and several built-in functions.

Directory Opus is a far cry from that class of directory utility. With features including unlimited files, up to 100 directories in memory at once, dozens of built-in functions, a completely configurable interface with comprehensive "online" help and full ARexx support, it is the most powerful directory utility available for the Amiga.

FILES AND DIRECTORIES

The Amiga's DOS (Disk Operating System) deals with two types of objects files and directories.

All files and directories must be given a unique name; you cannot have two files, two directories or a file and a directory with the same name in any given directory.

Files contain information, be it information from a database, from a word processor, a painting program, or a program itself. Any data you record on a disk are stored in a file.

Files have a definite size; this size is expressed in bytes. One byte is equivalent to one character. Therefore, to store the string "Hello" in a file would use take five bytes, as the word "Hello" is five characters long.

Files can also have other information associated with them.

The protection bits are a group of flags that are stored with the file, that determine the characteristics of the file. These flags are given one character names. The protection bits currently in use are HSPARWED.

H ... Hidden. If this flag is set, the file is not normally displayed. This allows you to mark certain files as "invisible", to avoid cluttering your directories. The file can still be accessed normally, and not all programs implement this flag.

S ... Script. A script file is a file containing a list of commands to run; it is like a simple computer program. This flag indicates that the file in question is a script file. A script file is sometimes called a batch file.

P ... Pure. If a program file is flagged as pure, it can be made to remain in memory, even when not in use. This can save a great deal of time, especially if the program is used a lot, as it does not have to be loaded from disk each time.

A ... Archive. This flag indicates that the file has not been changed. If this file is ever written to, the A flag will be turned off. This can be used in a hard disk backup program, to record which files have been backed-up, and need not be backed-up again.

R ... Read. If this flag is set, the file can be accessed.

W ... Write. If this flag is set, the file can be written to (ie, more information can be stored in it than is already there).

E ... Execute. If a program file does not have this flag set, it can not be run.

D ... Delete. If this flag is not set, the file can not be deleted.

All files have what is called a datestamp associated with them. This simply shows what the system date and time were the last time the file was written to.

Files may also have a comment attached to them. This is optional, and may be up to 79 characters long.

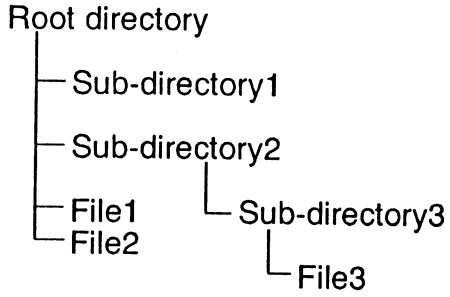
Directories are the other type of AmigaDOS object. If you picture a disk as a filing cabinet, with your programs, database files, pictures, etc. being the actual files themselves, then directories are analogous to the drawers of

the filing cabinet. Incidentally, directories are often called drawers.

This is a simple, but inaccurate analogy, however. While AmigaDOS directories can be straight forward, like a filing cabinet, you can also have directories within directories (these are called sub-directories), and directories within these directories, and so on, indefinitely.

The directory containing any given sub-directory is known as the sub-directory's parent directory.

There is also a very special directory : the root directory. If you imagine a disk as a filing cabinet, then the root directory is the room the filing cabinet is in. The root directory has no parent directory, but it can contain sub-directories and files. Examine the following diagram :



From this you can see that the root directory contains four objects : Sub-directory1, Sub-directory2, File1 and File2.

Sub-directory1 does not contain anything further; it is empty.

Sub-directory2, however, contains a further sub-directory, called Sub-directory3. Sub-directory3 contains a file called File3.

Therefore, Sub-directory2 is the parent of Sub-directory3.

File1, File2 and File3 may be pictures, programs, or any other type of file.

The number of files and sub-directories any given directory (including the root directory) can contain is only limited by the amount of space on the disk.

The route you take along a "directory tree" (as pictured above) to get to a file is called a path. Each branch of the tree along the path is separated

from the next branch and the previous branch by a / character (the divide sign below the question mark on the keyboard). For example, the path of Sub-directory3 (above) is Sub-directory2/Sub-directory3.

To refer to File3, you use what is called a pathname. That is, a path, followed by the name of a file. For example, Sub-directory2/Sub-directory3/File3.

THEORY

The theory behind a program such as Directory Opus (not that there are any other such programs!) is quite simple. Two windows are visible on the screen. These are called directory windows. Into each of these can be read a directory, from any device accessible by the Amiga. Files and/or directories can be selected in either of these windows, and the selected files/directories (sometimes referred to as entries) can be copied to the opposite window, deleted, renamed, or anything else that you feel like subjecting them to. Text files can be read, picture files can be viewed and sound files can be heard. Directory Opus offers much more than these "bare-bones" features, and you will learn more of this later.

INSTALLING DIRECTORY OPUS

Some people may want to always run Directory Opus from the distribution disk. If you are one of them, you may skip this section.

The easiest way to install Directory Opus is with the supplied installation program. This program is called, strangely enough, InstallOpus.

InstallOpus displays a list on the screen of the files you need to install. You may choose (by clicking on the check marks) not to install some or all of these files. Each file has a From and a To field associated with it. The From field defaults to the correct path on the distribution disk; it is unlikely you would ever want to change this.

The *Destination* field specifies where the file in question is to be copied. You may want to change this to suit yourself.

If you click on the Help gadget on the InstallOpus screen, and then click on one of the check gadgets, you will be presented with a brief description of the option you selected.

Choosing Okay at this point will start the installation procedure; choosing Cancel will abort it.

RUNNING DIRECTORY OPUS

The easiest way to run Directory Opus is from the Workbench. Simply double-click on its icon, and several seconds later, the screen will appear. Directory Opus can also be run from the CLI, and, more commonly, from your startup-sequence.

For those not in the know, the startup-sequence is a script file that is automatically executed when you first boot up the computer. It can contain instructions to do many different things; one of these being the running of Directory Opus.

To run Directory Opus from your startup-sequence you need first of all a text editor. A word processor (such as Notepad) may do equally well, but a text editor is best for this. A highly recommended commercial text editor is CygnusEd Professional, by ASDG-Inc. This is the text editor with which Directory Opus was written, and is a very powerful tool.

Run your text editor (or word processor), and load your startup-sequence into it. The startup-sequence resides in the S: directory of your boot disk. Pick a convenient place to place the new instruction. A good place would be the line immediately before the EndCLI instruction (at the end), but you can really put it anywhere. Add the line :

```
DirectoryOpus -i
```

That's all there is to it. There are a few points, though. Firstly, the above instruction assumes that Directory Opus is going to be found in the current path list. This is usually the current directory, the C: directory and the root directory of the boot disk. If Directory Opus is actually found somewhere else, the instruction will have to include the path as well. For instance, if the Directory Opus program was kept in the directory DH0:Tools/, and DH0:Tools/ was not in the current path list, the instruction would have to read :

```
DH0:Tools/DirectoryOpus -i
```

Secondly, the -i at the end of the instruction tells Directory Opus to start up

iconified. Briefly, iconified means that the program opens only a little window on the Workbench screen, instead of its own custom screen, and sits unobtrusively in the background until needed. The iconified state will be described in greater detail later. From the Workbench, you can add a ToolType of ICONSTART=1 to the Directory Opus icon, for the same effect. If you do not want Directory Opus to start up iconified, simply remove the -i. For instance :

```
DirectoryOpus
```

Thirdly, the more technically inclined may notice that there is no RUN on the instruction line. This is because Directory Opus detaches itself, freeing up the CLI and allowing it to continue with the startup-sequence (and ultimately closing).

Make sure you save the startup-sequence once you have modified it, and if all goes well, Directory Opus should be run automatically when you boot up.

There are also two other flags you can use when you run Directory Opus. The -c flag tells Directory Opus where to load the configuration file from. For instance, -cDH1:stuff/directoryopus.cfg would load (and subsequently save) the configuration file from DH1:stuff/directoryopus.cfg. From Workbench, a ToolType of CONFIGFILE=DH1:stuff/directoryopus.cfg would have the same effect.

The other flag, -o, lets you tell Directory Opus where to load the ConfigOpus (explained later) program from. For instance, -oDF2: would tell Directory Opus to load ConfigOpus from DF2:. Note that you may only specify a path with this flag; the program must still be called ConfigOpus. From Workbench, a ToolType of CONFIGPATH=DF2: would have the same effect.

TERMS

Some terms used in this manual that you may need to know are :

Active (directory window) ... denoted by a highlighted disk name. This is the source directory, the directory that all commands will act upon.

Buffers (directory) ... Directory Opus maintains up to 100 internal buffers (up to 50 for each directory window) of past directories. Buffer can

(in this case) also be used as a verb (ie, to buffer back and forth). To buffer back and forth, click on the arrow gadgets (this is described fully later on).

Click-M-Click ... this is a Directory Opus term. To click-m-click is to click once with the mouse button (again, usually the left button, but sometimes the right button), move the mouse to another part of the screen, and press the mouse button again, all within the double-click time set from Preferences. You will notice that the Directory Opus screen has a certain degree of symmetry, and you will usually move the mouse to the equivalent place on the other side of the screen for the second click.

Directory gadgets ... There are two rectangular boxes just underneath the horizontal proportional gadgets, at the bottom of the directory windows. These are the directory gadgets, and contain the pathname of the currently open directory.

Directory window ... one of the two windows on the screen in which entries are displayed.

Disk name ... the name of the disk on which the open directory resides is displayed above the directory window along with the amount of free space.

Double-click ... an Amiga term, this means to click twice with the mouse button (usually the left button, but sometimes the right button) on an object, within a pre-determined time. The time is set from within Preferences, and you should consult your Amiga manual for more information on this.

Drive gadgets ... the 24 gadgets, of which only six are visible at once, that enable you to easily read any directory.

Entry ... either a file or a directory.

File types ... a file type is a set of user-defined tests to see whether a file is of a certain type or not. Actions can then be performed on the file, depending on its type.

Gadget ... a well-used Amiga term. A gadget is simply a box on the screen, that does something when you click on it. A proportional, or slider gadget is one that allows you to select a value, by sliding a box up and down, or left and right (or sometimes both!). A string gadget is one that allows you to enter a string (or text), by selecting the gadget and typing as

normal.

Highlight ... highlight is the term given to the changing or reversal of colors, usually when an object is selected.

Inactive (directory window) ... disk name is not highlighted. This is the destination directory, and files may sometimes be written here from the source directory.

Menu ... a pull-down menu is like a list of gadgets. A menu is accessed by moving the mouse to the top of the screen and holding the right mouse button.

Open ... a directory that is currently displayed is said to be open.

Requester ... a requester is usually a box that appears on the screen, soliciting a response. Some requesters require a string input, others simply ask you to choose yes or no.

RMBGadget ... a Directory Opus term. This is a gadget that is selected with the right mouse button, instead of with the left (which is the more usual).

Selected ... entries are selected when they are highlighted. This is usually done by clicking on them with the left mouse button.

Status bar ... this is the bar at the very top of the Directory Opus screen, which displays information, error messages and the like.

Chapter 2:
Preliminary Information

ABORTING

Before we discuss any functions of the program, it is important to know how to abort an action. Most functions can be aborted once they have begun, and this can be useful if you accidentally start deleting the contents of your hard drive. To abort is quite simple; press the left and the right mouse buttons simultaneously. You should also note that this will only take effect if the Directory Opus screen is active (i.e., it is at the front of the display).

If the action does not abort immediately, do not be alarmed. Feel free to press the left and right mouse buttons as much as you like, but some functions, such as COPY, will have to finish the file they are working upon before aborting.

SELECTING THE ACTIVE DIRECTORY WINDOW

Before you perform any action, you must select the active directory window. Either window can be active (and therefore the source); it does not matter which one it is. The program starts up with the left window being active, but this can be easily changed in several different ways.

Clicking, with either the left or the right mouse button, anywhere in the window will activate it. The left and right buttons do, however, operate in slightly different ways. The left mouse button is used to select entries, and therefore, if you click with the left mouse button in order to activate either window, you may inadvertently select an entry you did not want selected. This is where the right mouse button comes in; clicking with the right mouse button will not select any entries as it activates the window.

You may also click with either mouse button on the disk name. This also has the added effect of updating the disk name display; should the name of the disk or the amount of free space been changed outside of Directory Opus, this will update it.

Another way to select the active window is by pressing the space bar. Many of the functions in Directory Opus have keyboard equivalents, and

this is one of them. Pressing the space bar will simply activate the non-active directory window. This allows you to toggle between the two.

There is another way to select the active window, and this is mentioned in the next section.

READING A DIRECTORY

Apart from the disk functions (detailed later), all functions require you to have read a directory into the directory window. There are several ways this can be accomplished.

At the bottom-left of the Directory Opus screen is a stack of six gadgets. These are the drive gadgets. There are actually 24 drive gadgets, but only six are visible at once. To view the next six, simple click the right mouse button anywhere over the drive gadget display (the entire drive gadget display is really a rmbgadget). These gadgets can be customized, as you will read later, but start off containing the names of all the devices and assigned directories present in your system.

As you have probably guessed, clicking on one of the drive gadgets will read that directory into the active directory window.

Another way to read any directory is to enter the name yourself. This process also sets the active directory window. Immediately above the forty eight gadgets at the bottom, and immediately below the two horizontal proportional gadgets, are two long boxes. These are actually string gadgets, and are called the directory gadgets.

If you click in either of these, a cursor will appear and you will be able to enter the name of the desired directory. Another way to activate the string gadget (of the active directory window) is to press the return key.

There are several editing features available in these (and other) string gadgets. Firstly, and obviously, pressing the backspace key will delete the character to the left of the cursor. Pressing the delete key will delete the character under the cursor.

Pressing the left and right cursor keys will move the cursor one space in that direction. Pressing these cursor keys while holding down either shift key will move the cursor to the beginning or end of the string. Finally, holding down the right Amiga key and pressing the x key (this is known as

AMIGA-X) will erase the entire string.

Once you have entered the pathname you wish to read, press return. You will see the directory window activate (if it is not already active), and the directory will start to read.

There are two other ways to read directories in. These are called the DEVICE LIST, and the DIRECTORY TREE, and are covered in further sections.

ENTERING SUB-DIRECTORIES

Many directories that you read in will contain sub-directories. Instead of having to type the name of a sub-directory in, should you wish to read it, you need only to double-click on it. This will load the sub-directory into the same window, advancing the buffer on by one.

You may also want the sub-directory to open in the other directory window, thus allowing you to copy files between the sub-directory and its parent. To do this, you need to click-m-click with the left mouse button, first on the sub-directory, and secondly on the other directory window.

MOVING THROUGH THE BUFFERS

Directory Opus maintains up to 50 buffers per directory window, enabling you to quickly access parent directories (or completely different directories). To move back and forth through the buffers, you can click on either of the buffer gadgets. The buffer gadgets at the inner ends of the directory gadgets. Clicking on either the left or right buffer gadget (on either directory window) will move one buffer in that direction.

The keyboard equivalent for the buffer gadgets is either ALT key in conjunction with the left or right cursor key.

MOVING THROUGH THE DIRECTORY LIST

Once the directory has been read in, it is displayed in the directory window. There is limited space on the screen, however, and so the whole directory may not be visible at once.

In the center of the two directory windows are two vertical proportional gadgets. You can slide these gadgets up and down, to determine the current position in the directory. To slide these gadgets, press the left mouse button on the solid bar, and while still holding the mouse button, move the mouse up and down. You may also click above or below the solid bar, in which case the slider will move one page (or part thereof) in that direction.

At the bottom of the two vertical sliders are arrow gadgets. These will move the slider one line in either direction when clicked upon. If you press and hold them with the left mouse button, they will repeat.

Alternatively, you can press either the up or down cursor keys, to move one line in that direction. These too will repeat. Pressing these keys in conjunction with either SHIFT key will move a page in that direction, and in conjunction with the CTRL key, will move to the top or bottom of the directory.

Whether or not the entire directory fits in vertically, it will definitely not fit in horizontally. There is more information available for entries than just the name (which, incidentally, is limited to thirty characters). Also displayed is the size in bytes, the protection bits assigned to this entry, the date that entry was last written to the disk, and any comment that may be assigned to the entry.

In a similar vein to the vertical proportional gadgets and arrows, horizontal proportional gadgets can be seen at the bottom of the directory windows, with arrows at the inner ends. These operate in a similar manner to their vertical counterparts, including the keyboard equivalents. Either cursor key will move one space in that directory, a page in conjunction with either SHIFT key, and to the beginning or end of the display in conjunction with the CTRL key.

An additional feature allows you to quickly scroll to the first entry starting with a particular letter. This is very useful when you are dealing with big directories. To perform this action, simple press the desired letter. The directory will scroll as close as it can get to the first entry starting with that letter.

Pressing a letter in conjunction with either SHIFT key will scroll only to the first file beginning with that letter. Pressing a letter without the SHIFT key will scroll to the first entry (directory or file).

You can also scroll up and down, and left and right, by holding the right mouse button and moving the mouse.

SELECTING ENTRIES

To select an entry, all you have to do is click on it with the left mouse button. If you hold the left mouse button and move the mouse up and down, all entries passed over will be selected. This is called drag-selecting.

There are other ways to select more than one entry, apart from drag-selecting. You can click on the ALL gadget, which will select all entries in the active directory window. Similarly, clicking on the NONE gadget will deselect all entries. Clicking on the ALL gadget with the right mouse button will reverse the states of all entries; that is, selected entries will be deselected and vice versa.

Between the directory gadgets you can see a gadget labelled S. This is the SELECT gadget, and it allows you to select single files or groups of files, based upon pattern matching.

Pattern matching is the term given to selecting entries based upon a specified pattern. For instance, it allows you select all entries that contain the letters "oon". The pattern matching provided by Directory Opus is very powerful, allowing you to select almost any combination of entries.

When the S gadget is clicked, you told to enter the select pattern. You can also choose, by clicking on the gadget marked All, whether the pattern will affect all entries, only files or only directories. The select pattern is a string consisting of normal characters, and any number of pattern matching commands. These commands are described below :

? matches any single character, but does not include none. For example, FI?E would match FILE, FINE and FIRE but not FIE.

matches any number of repetitions of a character, including none. For instance, FRE#D would match FRE, FRED, FREDD, FREDDD, etc.

() is in effect an or. (p1| p2| p3) will match on any one of the patterns p1, p2 and p3. For instance, (D*G| C*T) would match DOG, CAT, DIG, COT, etc. You may include as many patterns

inside the parenthesis as you like, separated by vertical bars.

% this will match on nothing. For example, CA(M| %)P will match CAMP and CAP.

* matches any number of repetitions of any characters, including none. For instance, F* will match F, FR, FRE, FRED, FREDD, FREDDY, etc..

~ this will NOT match on the following pattern. For example, F~(ROG) will match FROM but not FROG. ~J* will not match anything starting with J.

' the apostrophe means take the next character literally, not as a wildcard. For instance, YEAR*'% will match YEAR1989%, YEAR1990% but not YEAR1989 or YEAR1990 (as would normally be the case, with % meaning match on nothing). To match on an apostrophe use ''.

As you select entries, a count is displayed in the status bar, of the number of selected files, directories and bytes (total length of all selected files), out of the total number of files, directories and bytes (total length of all files). Above and to the right of the S gadget is a gadget marked R. This is the RESELECT gadget. This gadget will select all entries that were highlighted before the last function was activated. For instance, if you selected a group of files and copies them to another disk, selecting the R gadget would reselect the files that were highlighted. This gadget can also act as an undo feature for the ALL and NONE gadgets.

Chapter 3:
Default Gadgets

Directory Opus has 168 user-definable gadgets. The functions section, chapter 5, explains more about the gadgets, and chapter 6 explains how to configure them. This chapter simply gives a brief description of the default gadgets. All the gadgets can be changed, but they do come configured to run certain functions.

ALL

This selects all entries in the currently active directory window. Selecting ALL with the right mouse button will reverse the state of all entries; selected entries will be deselected and vice versa.

NONE

This deselects all entries in the currently active directory window.

PARENT

This will read the parent directory of the directory open in the active directory window. A configuration option, *Smart parent* (described later), will search all previous buffers for the parent before actually re-reading it. If the parent is found in a previous buffer, that buffer is displayed, which can prevent needless reading of disks.

There are also two hidden parent gadgets; one for each directory window. They are hidden in the outermost borders (on the left of the screen and the right of the screen) of the directory windows.

ROOT

This function reads the root directory of the active directory window. A configuration option, *Smart parent* (described later), will search all previous buffers for the root before actually re-reading it. If the root is found in a previous buffer, that buffer is displayed, which can prevent needless

reading of disks.

There are also two hidden root gadgets; one for each directory window. They are hidden in the outermost borders (on the left of the screen and the right of the screen) of the directory windows. With the left mouse button, these gadgets perform the parent function; they will load the root directory when selected with the right mouse button.

BYTE

This will display in the status bar the same information that is displayed when entries are selected (namely, the number of selected files, directories and bytes, out of the total number of files, directories and bytes), along with one additional piece of information. If the total selected bytes are less than the free space on the disk in the opposite directory window, and thus all selected files would fit on the destination disk were they to be copied, a Y will be displayed after the count. Otherwise, a N will be displayed.

If any directories are selected when BYTE is chosen, the directories that do not already have a size displayed (see below) are scanned, and the size of the directory is calculated and displayed.

If the size of a directory is already displayed, then the directory will not be re-scanned even if it is selected and BYTE is chosen. This could mean that the directory size is inaccurate, especially if any files have been created or deleted in it by another program. Selecting one or more directories, and then clicking on BYTE with the right mouse button (it is also an rmbgadget), will cause the directory sizes (if displayed) to disappear. You can then reselect the directories, choose BYTE (with the left mouse button), and they will be re-scanned.

When any complete operation is performed on a directory (e.g., COPY, PROTECT, HUNT, etc.), the total number of bytes in that directory (ie the sum of the lengths of all files in that directory and its subdirectories) is displayed as the size of that directory. This only occurs after a complete operation has been performed; if the operation is aborted midstream (by operator or by error), the size is not shown.

MAKEDIR

This function allows you to create a new sub-directory in the active directory window. A configuration option, *Create icons with directories*, will cause an icon to be created at the same time. The name of the directory is limited to 30 characters, or 25 if Create icons with directories is enabled.

DELETE

This will delete all selected entries in the active directory window. Be careful with this, as it is easy to wipe out valuable data if you are careless.

If the configuration option *Ask before commencing delete* is enabled, you will be asked to verify your choice before any deleting commences.

If the configuration option *Ask before deleting files* is enabled, you will be asked for confirmation before each file is deleted.

If the configuration option *Ask before deleting non-empty directories* is selected, you will be asked for confirmation before every non-empty directory is deleted.

The configuration option *Set delete protection bit* will cause all files that are protected against deletion to be deleted anyway.

CLONE

This function allows you to make a copy of selected entries in the same directory, but with different names. A requester will appear for each entry, asking for the new name.

COPY

This is similar to CLONE except that the entries are copied to the destination directory (inactive directory window), keeping their original names.

RENAME

This function allows you to give new names to all selected entries in the active directory window. A requester will appear for each entry in turn, asking for the new name. The initial rename requester has two string gadgets instead of one. You will usually just edit the name in the lower of the two to the new name. You may, however, perform a limited type of wildcard rename.

Entering a * in the bottom gadget allows you to add prefixes or suffixes. For instance, entering *.pic will add a .pic suffix to all selected entries. Entering A* will add an A prefix. Only one * may be used in this process, and this also means that you cannot give a file a new name that contains a *.

If you enter a * in the top as well as the bottom gadget, you can replace sections of the name. For example, entering *.pic in the top gadget and *.iff in the bottom gadget will replace the .pic suffix in any entry that has one with a .iff suffix. If an entry does not have a .pic suffix, it will be left untouched. The * may also be embedded. For instance, renaming FOO*BAZ as GEE*WIZ would rename FOOBARBAZ as GEEBARWIZ. Again, only one * may be used in each of the string gadgets.

MOVE

This function will move all selected entries from the source directory to the destination directory. The entry will no longer exist in its original place.

If this gadget is selected with the right mouse button, you will be able to give each entry a new name before it is moved.

EDIT

This function allows you to edit any selected files, using the text editor (or word processor) specified. This gadget is initially configured to call the program c:Editor as the editor. You will probably wish to change this; information on how to edit the gadgets is contained in chapter 6.

If you select this gadget with the right mouse button, you will be asked for a filename to create a new file.

RUN

This function allows you to run each selected file in turn, providing that file is executable. This will have much the same effect as if you had double-clicked on the file's icon, or run it from the CLI. A requester will appear, asking for any arguments (should you require any).

If you select this gadget with the right mouse button, the files will instead be treated as batch files, and executed. This has the same effect as if you had run the batch file via IconX, or executed it from the CLI.

If no files are selected when this gadget is chosen, Directory Opus will open a CLI window. This is a normal CLI, and you must type endcli within it to remove it.

ENCRYPT

Ever had files that you wanted to encrypt so that only people who knew the password could understand them? ENCRYPT allows you to do just that. This function will encrypt all selected files around a password that you enter. The resulting files are not written over the originals, but are instead written to the destination directory. They will be the same size as the original files, so you can ensure you have enough room in the destination directory.

The files are encrypted, using the password you enter, with a complex algorithm that most people will find impossible to work out.

To decrypt a previously encrypted file, you should enter the same password preceded by a minus sign. For example, to decrypt files you encrypted with the password FOO, select the files, choose ENCRYPT and enter -FOO as the password.

PLAY

This will attempt to play each selected file. If the file is an 8SVX format IFF sampled sound (of the type generated by AudioMaster, for instance), the speed and other information from within the file will be used. This will play any file, though, and if the file is not 8SVX, it will assume a speed of 10000 samples/second.

This function also plays SoundTracker modules, provided the stopus.library file is present in the LIBS: directory. Modules are recognised by the mod. prefix in the filename; if the filename does not have this prefix, it will not be played.

Each file is played once only. To abort a sound while it is playing, press the left and right mouse buttons.

If you select this gadget with the right mouse button, the sounds are played continuously (looped). To advance to the next sound, you must press the left mouse button.

SHOW

This function is very versatile indeed. It will display IFF ILBM pictures and brushes, Workbench icons and fonts.

If the file is an IFF ILBM, it will be displayed using information from the file. Directory Opus will correctly show most pictures and brushes, including overscan, extra halfbrite (EHB), HAM (4096 colour) pictures, and Dynamic HiRes pictures created by MacroPaint (from Lake Forest Logic).

If the picture contains color cycling information, pressing TAB will toggle colour cycling on or off. You can press P to print the picture. Should you wish the mouse pointer to be visible, in order to point at some part of the picture for instance, press the . key.

The show routine will also display icons. If the file has a .info suffix it is assumed to be an icon. If the icon has an alternate image, pressing return will toggle between the two.

To display a font, you need to enter the actual font drawer, and show the size file. That is, the file 8, 12, 19, etc., NOT the .font file.

READ

This function allows you to read selected files. At the bottom of the screen the name of the file is displayed, along with the current position within the file in lines and as a percentage, and the also the total number of lines in the file.

The mouse is used to move through the file. Press the left mouse button to turn scrolling on or off. The direction and speed of the scroll are governed by the position of the mouse pointer in the window. The mouse pointer will become invisible while the text is scrolling.

No scrolling occurs if you have the mouse in the center of the window. To scroll forward, move the mouse down until the text starts to scroll (providing there is actually more than one page of text). The further down you move the mouse, the faster the text will scroll. This procedure is reversed for backwards scrolling.

The gadgets in the bottom-right of the screen also allow you to move around the file. Most gadgets will repeat if held down for any length of time.

The up and down arrows move up and down a line at a time. The U and D gadgets move up and down a page at a time. The T and B gadgets move to the top and bottom of the file.

You can also use the cursor keys to move up and down. As before, in conjunction with SHIFT these will move a page at a time, and in conjunction with CTRL will move to the top and bottom.

To jump to a specified line, click on the number showing the current line position (immediately left of the word Lines). You can also press the J key. To jump to a specified percentage, click on the number immediately left of the % sign.

To search for a string, click on the S gadget or press the S key. You can use full pattern matching in this search.

To print the current file, click on the P gadget (or press the P key). To print only the current page, press the C key (this has no gadget equivalent).

To leave the text viewer, click on the X gadget, or press either X, Q or ESC.

If more than one file were selected, the next one will be read when you exit. To exit without reading the next file, press the right mouse button.

HEX READ

This function will read the selected files in the same way as READ, except in hexadecimal format. This allows you to view binary and other files containing non-text characters. The file is displayed in the following manner :

```
00000000: 000003F3 00000000 00000001 00000000 ...ó.....
00000010: 00000000 0000013F 000003E9 0000013F .....?...é...?
00000020: 6002B104 00000F60 00000000 48E78080 '.Jü.....Hç..
```

The first value is the offset, displayed in hex. This is the number of bytes you are into the file. The next four values are each a four byte longword, with the actual ASCII representation at the end. Any non-text characters are shown as a . character.

ADD ICON

This function allows you to add icons to all selected entries in the active directory window. Directory Opus will automatically sense what type of file it is and add the appropriate icon (drawer, tool or project). Default icons are used unless you have specified the name of your own icons within the configuration (described later).

ASSIGN

This function allows you to assign a logical device name to the currently active directory. The C: device is an example of a logical device; it is not a physical disk itself, but is a directory that has been assigned the name C:.

For example, a simple name such as TEXT: can be assigned to a long and complicated pathname such as

DH0:Projects/Work/1989/April/Bits&Pieces/Text

From then on, when you entered TEXT: into a directory gadget, you will go straight to the directory at the end of the assigned path, instead of having to go through all sub-directories on the way. Logical devices always end with a colon.

ASSIGN also allows you to cancel a previously assigned logical device, or

reassign a device to another directory. To do this, press the right mouse button over the S (select) gadget to activate the DEVICE LIST function. Select the assigned device(s) you wish to change, and click on the ASSIGN gadget. A requester will appear, into which you can type the new path name to be assigned that device name. If you enter a blank string, that logical device will be unassigned. All selected devices will be unassigned if you enter a blank string. Otherwise, you will be prompted for a new path name for each selected device.

PRINT

The print routine is a full-featured text formatter. There are several configuration items in the print requester :

Page length: This is the length of each page, in lines.
Left margin: This is the position of the left margin, in characters.
Right margin: This is the position of the right margin.
Tab: This is the number of spaces a tab character is equivalent to.

You can have a header and/or a footer printed at the top and bottom of each page if you wish. Both the header and footer can consist of the title, the current date and the page number. You may select any combination of Title, Date and Page # you wish. If none are selected, the header or footer is not printed.

You may also select the typestyle of the header and the footer, by clicking on the gadgets that initially say Normal. These allow you to cycle through the various typestyles available : Normal, Bold, Italics, Underline, Doublestrike and Shadow. Note that some printers may not have all these typestyles available.

You may enter a title (to be used in the header and footer) in the Title string gadget. Leaving this blank will use the name of the file as a title.

The Output gadget allows you to specify where you want the output of the print process to go to. This defaults to PRT:, but you may choose to have the output sent to a disk file. In this case, simply enter the path and the name of the file you wish to print to.

The gadget marked *Normal* at the bottom of the print requester operates in the same way as the header and footer typestyle gadgets, except that it determines the typestyle for all text other than headers and footers.

The gadget marked *Draft* allows you to specify whether all text is printed in draft mode or NLQ mode, headers and footers are printed in draft mode with the rest of the text in NLQ, or headers and footers in NLQ with the rest of the text in draft.

The gadget next to this one determines whether or not the last page of the file to be printed is followed by a formfeed character, ejecting it.

To start the print operation, select the *Print* gadget.

This function will print all selected files, one at a time. If you select only one file to print, the print routine will be started up as a separate process, allowing you to continue working with Directory Opus. To cancel this type of print, simply select the print function again. A requester will appear asking if you want to continue with the print or halt it. This requester will also appear if you attempt to quit Directory Opus while a print operation is in place, as you cannot quit until the print has finished.

To cancel a print that has not been spooled, simply press the left and right mouse buttons in the normal way.

Note that even if you abort a print, the printer may not actually stop for sometime. This is because most printers have buffers, some quite large ones, which store data faster than it can be printed.

ARCHIVE

This gadget allows you to list and extract files from archives created by LHArc, Arc and Zoo (and other archives as well; see the File types section of chapter 6 for more information).

To list one or more archives, select the files and click this gadget with the left mouse button.

To extract files from one or more archives, select the archives and click this gadget with the right mouse button. You should have the destination directory open in the inactive window.

ARCHIVE A

This gadget allows you to create and add files to archives, using the

popular archivers Arc, Zoo and LHArc (and up to three others; see the Functions section of chapter 6 for more information).

To add files to an archive, select the files you wish to add, and click on this gadget. You will be asked for the name of the archive to create or add to. You should have the destination directory open in the inactive window.

COMMENT

This function allows you to add comments to all selected entries, or edit existing comments. The maximum length of a comment is 79 characters. If any directories are selected to comment, you are asked if you wish to comment the directories recursively. If you answer yes to this, all files and sub-directories within that directory will be commented as well. If not, only the directory itself will be commented.

You will be prompted for a comment for each selected entry. If you wish to attach the same comment to all entries, you should select the All gadget.

DATESTAMP

This function allows you to change the datestamp of the selected files and directories in the active directory window. If any directories are selected, you are asked whether you wish the files within them to have their datestamps modified also.

For each entry, you are presented with a requester. If you wish the file to have its datestamp set to the current date and time, simply press return. Otherwise, enter the desired date and time.

To set the datestamp of all selected entries you should select the *All* gadget from the datestamp requester; choosing *Okay* or pressing return will set the datestamps a file at a time.

PROTECT

This function allows you to modify the protection bits of the selected files and directories in the active directory window. If any directories are selected, you are asked whether you wish the files within them to be protected as well.

For each entry, you are presented with a requester displaying the protection bits currently set for that entry. The protection bits are **H** (hidden), **S** (script), **P** (pure), **A** (archived), **R** (readable), **W** (writable), **E** (executable) and **D** (deleteable).

To toggle bits on or off, click on the appropriate gadgets, or press the appropriate keys.

To set the protection bits of all selected entries you should select the *All* gadget from the protection requester; choosing *Okay* will set the protection bits a file at a time.

HUNT

This function allows you to search all selected directories (and their sub-directories) for a specified file or files. A requester appears asking for the pattern to hunt for. You can use full pattern matching for this search.

If a file matching the pattern is found, you are asked if you wish to enter the directory containing it, or continue the search. If you elect to enter the directory, the directory will be read and then all matching entries will be highlighted.

You can also select entire devices to hunt, using the DEVICE LIST feature that is described later.

SEARCH

This function allows you to search all selected files, and the files within selected directories, for a specified string. A requester appears asking for the string pattern to hunt for. Again, full pattern matching is supported.

If a file containing the string is found, you are asked if you wish to read that file, or continue the search. If you elect to read the file, it will be loaded into the text viewer, and a search will be automatically initiated for the desired string.

You can also select entire devices to hunt, using the DEVICE LIST feature that is described later.

THE TINY GADGETS

The Directory Opus screen also has several other gadgets on it. These are not configurable; they activate the very useful or necessary functions, and are included so that no matter what your configuration, these functions are always available. For instance, if you had Directory Opus configured so that no gadget or menu called the configuration program, you would never be able to run the configuration program to change this! A gadget is provided, however, to call the configuration program no matter what your setup.

These gadgets are called the tiny gadgets, as they are much smaller than the configurable gadgets on the screen. There are four of them arranged in a square in approximately the center of the screen. There are also five in the bottom right-hand corner of the screen. They all have one character titles.

B - Buffer list

This activates the bufferlist function. This function displays a list of all the directories contained in Directory Opus' internal buffers. You may then double-click on one of the displayed buffers to jump to that buffer immediately, rather than clicking the arrows to cycle through the buffers one by one.

Pressing the right mouse button over this gadget will activate the clearbuffers command. This will clear from memory the contents of all buffers except the two that are currently displayed.

R - Reselect

This activates the reselect function. This function will reselect all entries that were selected before the last operation was initiated. The entries are reselected only if they still exist, and even if the buffer containing them is not displayed currently.

Pressing the right mouse button over this gadget performs the rescan function on the active directory window.

S - Select

This gadget activates the select function. This allows you to select files and directories in the active directory window using wildcards. All standard wildcards are supported (*,~,? etc..) as well as the AmigaDOS wildcards #? and ?.

Pressing the right mouse button over this gadget will activate the getdevices function (the DEVICE LIST function). This function produces a list of all devices, volumes and logically assigned directories that exist in the system.

A - ARexx

This activates the arexx function. This function allows you to launch an ARexx script or execute an ARexx command from within Directory Opus.

Pressing the right mouse button over this gadget will activate the dirtree function (the DIRECTORY TREE function). This function scans all sub-directories in the currently displayed directory, and produces a tree structure.

? - Help

This gadget activates the help function. Provided a help file has been loaded in, selecting any gadget or menu after enabling the help mode should result in the appearance of some helpful text.

E - Error help

This activates the errorhelp function. This function gives you more information about DOS error codes than is generally available.

C - Configure

This activates the configure function, and will either load into memory or invoke the configuration program.

I - Iconify

This gadget activates the iconify function. It closes the Directory Opus screen, deallocates as much memory as possible, and then opens a small window on the Workbench screen.

This allows you to have Directory Opus constantly available, while using the minimum amount of memory possible.

Q - Quit

This activates the quit function. This function will exit Directory Opus (providing there is not a print operation happening at the moment). If the configuration has been modified and not saved, you are asked if you wish to save it first.

Chapter 4:
Default Menus

Directory Opus has 100 user-definable menus. The functions section, chapter 5, explains more about the menus, and chapter 6 explains how to configure them. This chapter simply gives a brief description of the default menus. All the menus can be changed, but they do come configured to run certain functions.

Project/Set date

This function allows you to set the system time and date, especially if you do not have a battery-backed clock connected. After you have set the time, the Amiga clock keeps running to provide an accurate timekeeper. All files or directories created will have a datestamp containing the current system time. The DATESTAMP function of Directory Opus also uses the system time.

The requesters will appear; the first containing the current date, and the second containing the current time. To change either the date or the time, delete and enter a new one. Date is expected in the form DD-MMM-YY. For example, 22-Sep-89. Time is expected in the form HH:MM:SS, as in 18:30:43. You must include the dashes and colons as demonstrated above.

Project/Current dir

This function allows you to set the current working directory for Directory Opus. The current directory is the directory loaded when you activate either directory gadget, delete the existing pathname (if there is one), and press return on a blank string.

A requester will appear, containing the pathname of the active directory window. If this is acceptable as a new current directory, just press return. Otherwise, edit it to read as you wish.

Project/Help!

Provided a help file has been loaded in, selecting any gadget/menu after enabling help mode with this menu should result in the appearance of some helpful text.

Help can be read from any file, although the default is S:DirectoryOpus.HLP. The file name can be changed with the Helpfile configuration option, described later.

Chapter 8 describes the help file in detail, and explains how you can edit it to add help for your own gadgets and menus.

Project/Error help

Error help gives you more information about DOS error codes than is generally available. When you select this option, you are asked for the DOS error code you desire help with. Examples of DOS error codes are 123, 205 and 225. You are then presented with a description of that error, and information pertaining to the possible cause and cure of the error.

Error help is also stored in the help file (see above); if the help file has not been loaded, error help will not be available. See chapter 8 for information on the help file.

Project/Configure

Directory Opus uses a separate program to modify the configuration. This results in a saving of approximately 70K of memory, as the configuration program need only be in memory when it is being used. As you will very rarely need to call upon the configuration program once you have your configuration finalized, this can be a great advantage.

If you do have plenty of memory, however, Directory Opus can be configured to load the configuration program on start-up. In this case, it is run from memory each time the configure function is invoked, rather than loaded from disk. This can be much faster, especially if you have a floppy-based system.

See chapter 6 for detailed information about the configuration program.

Project/About

This function displays some information about the program, including the current version number, copyright information, and how to contact technical support.

Project/Version

This menu displays in the status bar the current version of Kickstart, Workbench and Directory Opus.

Project/Iconify

This function closes the Directory Opus window and screen, deallocates as much memory as possible, and then opens a small window on the Workbench screen. This is known as iconifying, and allows you to have Directory Opus constantly available, while using the minimum amount of memory.

The iconfied window may contain a clock, or not be visible at all, depending on how you have the Iconify type configuration option set (described later).

To determine the initial position of the iconified window (especially if you start Directory Opus up in the iconified state), you should iconify Directory Opus, and position the window as you would like it to appear. Then, re-enter Directory Opus, and save the configuration (described later).

To re-enter Directory Opus, simple click the left mouse button in the iconified window, then press the right mouse button. If you wish to quit Directory Opus without going back into it, just click the close gadget at the far left of the iconified window.

Note that if you have the configuration option Iconify type set to hidden, the only way to re-enter Directory Opus is with the hotkeys. These default to CTRL-SHIFT-ALT (the CTRL key, the left SHIFT key and the left ALT key simultaneously), although this may be changed in the configuration (explained later).

Project/Quit

This function will exit Directory Opus (providing there is not a print operation happening at the moment). If the configuration has been modified and not saved, you are asked if you wish to save it first.

Disk/Disk copy

This function allows you to make an exact copy of one disk on another. When this function is invoked, a requester appears with several gadgets.

The *Source* and *Destination* gadgets can be clicked on with the left mouse button to select the drives (0-3). If you only have one drive, you can select DF0: as both source and destination, and you will be prompted to swap disks when necessary.

If *Verify* is selected, the copy will be verified after writing, to check for any errors. Verify turned off is faster, but will not pick up any errors.

This function will not copy any protected software, or non-AmigaDOS format disks.

Disk/Format

This allows you to format a new disk. All new disks need to be formatted before the computer can write to them.

When this is selected, a requester appears with several gadgets. Drives DF0: through DF3: are available. Seperate names may be specified for each disk.

Verify allows you to turn verify on or off. If verify is turned off, the format process will be much faster, but you will not be made aware of any errors that occur. If you don't completely trust your disks, leave verify on.

If *Quick* is selected, the disk will just be initialized (wiped). This provides an extremely fast way to erase an old disk. This will not work on new disks, however; only on disks that have previously been formatted.

If *No icons* is selected, the Disk.info, Trashcan.info and Trashcan directory will not be created on the disk.

Disk/Install

This function allows you to make a disk bootable. All bootable disks have special information written on the first sector of the disk. Without this, the disk will not boot.

DF0:, DF1:, DF2, and/or *DF3:* may be selected for installation.

If the *Check* gadget is selected, the disk will not be installed. Instead, the already existing bootblock will be compared with the standard one. If the bootblock is non-standard, you may want to run a virus checker program to check for the possible presence of a virus.

Disk/Relabel

This function allows you to change the name (label) of the disk that is open in the active directory window.

Disk/Print dir

This function allows you to print the directory in the active directory window. You are able to choose what information you want to print (file sizes, protection bits, dates and comments). This function is not started as a separate process, unlike the print command.

Disk/Disk info

This function displays some information about the disk the active directory resides on, including space used and free, datestamp and number of errors on the disk.

Disk/Add buffers

This function allows you to add cache buffers to the drive currently open in the active directory window. Each cache buffer takes approximately 512 byte of memory. Cache buffers speed up disk access considerably, especially for floppy drives. Some hard drives may not be sped up by cache buffers, and RAM drives and some hard drives may actually be slowed down.

Chapter 5:
Functions

Directory Opus takes a completely new approach to functions. Directory Opus has 84 gadgets (actually 168, but we will come to that later) and 100 menus that can be completely customized. Although the default configuration has 28 gadgets and 15 menus set up to perform internal functions, these can all be changed or completely removed at the user's discretion.

Chapter 6 gives complete details on how to configure the gadgets and menus; this section lists the internal functions that can be tied to any gadget or menu.

Note that under the default configuration, several gadgets have right mouse button functions as well. These too are configurable, and also call internal functions.

The function names listed below are the actual names of the functions; if they are, by default, tied to a gadget or menu, that gadget or menu is listed to the right of the function name.

abortprint

This function aborts the current print process (if one exists).

about Project/About menu

This function displays the about requester, which contains the current version number, copyright information, and tells how to contact technical support.

addbuffers Disk/Add buffers menu

This function allows you to add cache buffers to the drive currently open in the active directory window. Each cache buffer takes approximately 512 bytes of memory. Cache buffers speed up disk access considerably,

especially for floppy drives. Some hard drives may not be sped up by cache buffers, and RAM drives and some hard drives may actually be slowed down.

addicon ADD ICON gadget

This will add icons to all selected files and directories in the active directory window. Directory Opus automatically senses what sort of file it is (drawer, project or tool) and will add the appropriate icon.

all ALL gadget

This function selects all files and directories in the active directory window.

archive ARCHIVE A gadget

This function allows you to create and add files to archives. The default configuration allows the usage of Arc, Zoo and LHArc to create archives. You may modify these, or add up to three more archive types.

To add files to an archive, open the directory you wish the archive to be written to in the inactive window, select the file or directory (or files and directories), and click this gadget. You will be asked for the name of the archive to be created or added to.

Note that Arc, unlike Zoo and LHArc, does not allow you to archive directories.

arexx A gadget

This allows you to launch an ARexx script or execute an ARexx command from within Directory Opus (described later).

assign ASSIGN gadget

This function allows you to assign a logical device name to the currently active directory. The C: device is an example of a logical device; it is not a physical disk itself, but is a directory that has been assigned the name C:.

For example, a simple name such as TEXT: can be assigned to a long and complicated pathname such as :

DH0:Projects/Work/1989/April/Bits&Pieces/Text

From then on, when you entered TEXT: into a directory gadget, you would go straight to the directory at the end of the assigned path, instead of having to go through all sub-directories on the way. Logical devices always end with a colon.

Assign also allows you to cancel a previously assigned logical device, or reassign a device to another directory. To do this, press the right mouse button over the S (select) gadget to activate the DEVICE LIST function. Select the assigned device(s) you wish to change, and invoke the assign function. A requester will appear, into which you can type the new path name to be assigned that device name. If you enter a blank string, that logical device will be unassigned. All selected devices will be unassigned if you enter a blank string. Otherwise, you will be prompted for a new path name for each selected device.

auto ARCHIVE gadget

Directory Opus allows you to define your own file types, and specify what happens to a file of that type under four different circumstances. The auto and auto2 functions operate on file types. File types are explained fully in a later section.

By default, the ARCHIVE gadget calls the auto function (and the auto2 function via the right mouse button), to list and extract Arc, Zoo and LHArc archives.

Therefore, if you select an archived file and click on the ARCHIVE gadget with the left mouse button, the contents of the archive will be listed.

auto2 ARCHIVE gadget (rmb)

Directory Opus allows you to define your own file types, and specify what happens to a file of that type under four different circumstances. The auto and auto2 functions operate on file types. File types are explained fully in a later section.

By default, the ARCHIVE gadget calls the auto function (and the auto2 function via the right mouse button), to list and extract Arc, Zoo and LHArc archives.

Therefore, if you select an archived file and click on the ARCHIVE gadget with the right mouse button, the contents of the archive will be extracted to the destination directory.

bufferlist B gadget

This function displays a list of all the directories contained in Directory Opus' internal buffers. You may then double-click on one of the displayed buffers to jump to that buffer immediately, rather than clicking the arrows to cycle through the buffers one by one.

byte BYTE gadget

This will display in the status bar the same information that is displayed when entries are selected (namely, the number of selected files, directories and bytes, out of the total number of files, directories and bytes), along with one additional piece of information. If the selected bytes is less than the free space on the disk in the opposite directory window, and thus all selected files would fit on the destination disk were they to be copied, a Y will be displayed after the count. Otherwise, a N will be display.

If any directories are selected when BYTE is chosen, the directories that do not already have a size displayed are scanned, and the size of the directory is calculated and displayed.

cd Project/Current dir menu

This function allows you to set the current working directory for Directory Opus. This is the directory that is read if you press return on an empty directory gadget.

clearbuffers B gadget (rmb)

This function will clear the contents of all buffers other than the two that are currently displayed. All used memory will be deallocated; this is a good

way to free up memory quickly if you have lots of used buffers and are running low.

clone CLONE gadget

This function allows you to make a copy of selected entries in the same directory, but with different names. A requester will appear for each entry, asking for the new name.

comment COMMENT gadget

This function allows you to add comments to all selected entries, or edit existing comments. The maximum length of a comment is 79 characters. If any directories are selected to comment, you are asked if you wish to comment the directories recursively. If you answer yes to this, all files and sub-directories within that directory will be commented as well. If not, only the directory itself will be commented.

You will be prompted for a comment for each selected entry. If you wish to attach the same comment to all entries, you should select the All gadget.

configure Project/Configure menu

Directory Opus uses a separate program to modify the configuration. This results in a saving of approximately 70K of memory, as the configuration program need only be in memory when it is being used. As you will very rarely need to call upon the configuration program once you have your configuration finalized, this can be a great advantage.

If you do have plenty of memory, however, Directory Opus can be configured to load the configuration program on start-up. In this case, it is run from memory each time the configure function is invoked, rather than loaded from disk. This can be much faster, especially if you have a floppy-based system.

contst

This command will continue playing any SoundTracker module that has been stopped via the **stopst** command. This applies only if you are using

the stopus.library, not the streplay.library.

copy COPY gadget

This function copies all selected files and directories in the active directory window to the destination window, keeping the same names.

date Project/Set date menu

This function allows you to set the system time and date. You may wish to do this if you do not have a battery-backed clock connected.

datestamp DATESTAMP gadget

This function allows you to change the datestamp of the selected files and directories in the active directory window. If any directories are selected, you are asked whether you wish the files within them to have their datestamps modified also.

For each entry, you are presented with a requester. If you wish the file to have its datestamp set to the current date and time, simply press return. Otherwise, enter the desired date and time.

To set the datestamp of all selected entries you should select the All gadget from the datestamp requester; choosing Okay will set the datestamps a file at a time.

default

This causes Directory Opus to reset to the hard-coded, default configuration. If you have modified the configuration yourself, and have not yet saved it, you are asked if you wish to save it first.

delete DELETE gadget

This will delete all selected entries in the active directory window. Be careful with this, as it is easy to wipe out valuable data if you are careless.

dirtree A gadget (rmb)

This function generates a list of all sub-directories in the directory displayed in the active window, in tree format. You may instantly read any sub-directory by double-clicking on its entry in the tree, rather than travelling there along via all the prior directories.

diskchange

This command causes a diskchange event to be generated for the disk open in the active directory window

diskcopy Disk/Disk copy menu

This function allows you to make an exact copy of one disk on another. When this function is invoked, a requester appears with several gadgets.

The Source and Destination gadgets (DF0:, DF1:, DF2:, DF3:)can be clicked on with the left mouse button . If you only have one drive, you can select DF0: as both source and destination, and you will be prompted to swap disks when necessary.

If Verify is selected, the copy will be verified after writing, to check for any errors. Verify turned off is faster, but will not pick up any errors. If you are copying a disk to multiple destinations, with verify selected, and an error occurs, you will asked if you want to retry, remove or abort. Retry will rewrite the track and check for errors again. Abort will abort the entire copy procedure. Remove will remove that disk form the copy procedure. and Opus will continue to copy to the remaining disks. This feature prevents, for instance, a diskcopy to four disks being aborted at track 78 because of an error on just one of the destination disks.

This function will not copy any protected software, or non-AmigaDOS format disks.

diskinfo Disk/Disk info menu

This function displays some information about the disk the active directory resides on, including space used and free, datestamp and number of errors on the disk.

encrypt ENCRYPT gadget

Ever had files that you wanted to encrypt so that only people who knew the password could understand them? This function allows you to do just that. It will encrypt all selected files around a password that you enter. The resulting files are not written over the originals, but are instead written to the destination directory. They will be the same size as the original files, so you can ensure you have enough room in the destination directory.

The files are encrypted, using the password you enter, with a complex algorithm that most people will find impossible to work out.

To decrypt a previously encrypted file, you should enter the same password preceded by a minus sign. For example, to decrypt files you encrypted with the password FOO, select the files, choose the encrypt function and enter -FOO as the password.

errorhelp Project/Error help menu

This function gives you more information about DOS error codes than is generally available. When you select this option, you are asked for the DOS error code you desire help with. Examples of DOS error codes are 123, 205 and 225. You are then presented with a description of that error, and information pertaining to the possible cause and cure of the error.

This help information is stored in the help file - you may edit it or remove it as you wish.

execute RUN gadget (rmb)

This function will attempt to execute each selected file in turn as a batch file. This has the same effect as if you had run the batch file via IconX, or via the Execute command from the CLI.

format Disk/Format menu

This allows you to format a new disk. All new disks need to be formatted before the computer can write to them.

When this is selected, you are presented with four gadgets labelled DF0:,

DF1:, DF2:, and DF3:. Select or deselect these depending on which disks you want to format. Next to these gadgets, you may enter the name you wish to give each formatted disk.

Verify allows you to turn verify on or off. If verify is turned off, the format process will be much faster, but you will not be made aware of any errors that occur. If you don't completely trust your disks, leave verify on.

If Quick is selected, the disk will just be initialized (wiped). This provides an extremely fast way to erase an old disk. This will not work on new disks, however; only on disks that have previously been formatted.

If No icons is selected, Disk.info, the disk icon will not be created on the disk. However, Trashcan.info and Trashcan directory will be created as usual.

The FFS option in Format allows you to format a disk to use the Fast Filing System of Kickstart 2.0. If you are running a Kickstart less than 2.0, you will still be able to select this option. However, you will not be able to read FFS-formatted disks without 2.0 or greater.

As with **diskcopy**, when formatting multiple disks with the verify option, if an error occurs on one of the disks being formatted, you are given the option to continue, abort the entire procedure, or simply remove the offending disk (allowing the other disks to continue).

getdevices S gadget (rmb)

This function displays a list of all devices, volumes and assigned directories present in the system. You may then read these devices in by double-clicking on them. You can also select entire devices for use with the hunt and search functions.

help Project/Help! menu

Provided a help file has been loaded in, selecting any gadget/menu after enabling help mode with this menu should result in the appearance of some helpful text. You may edit the help file yourself; this is described later. To turn help mode off, select this function again.

hexread HEX READ gadget

This function will read the selected files in the same way as READ, except in hexadecimal format. This allows you to view binary and other files containing non-text characters. The file is displayed in the following manner:

```
00000000: 000003F3 00000000 00000001 00000000 ...ó.....
00000010: 00000000 0000013F 000003E9 0000013F .......?...é...?
00000020: 6002103F 00000FA0 00000000 48E78080 '.Jü.....Hç...
```

The first value is the offset, displayed in hex. This is the number of bytes you are into the file. The next four values are each a four byte longword, with the actual ASCII representation at the end. Any non-text characters are shown as a . character.

hunt HUNT gadget

This function allows you to search all selected directories (and their sub-directories) for a specified file or files. A requester appears asking for the pattern to hunt for. You can use full pattern matching for this search.

If a file matching the pattern is found, you are asked if you wish to enter the directory containing it, or continue the search. If you elect to enter the directory, the directory will be read and then all matching entries will be highlighted.

iconify Project/Iconify menu

This function closes the Directory Opus window and screen, deallocates as much memory as possible, and then opens a small window on the Workbench screen. This is known as iconifying, and allows you to have Directory Opus constantly available, while using the minimum amount of memory.

To determine the initial position of the iconified window (especially if you start Directory Opus up in the iconified state), you should iconify Directory Opus, and position the window as you would like it to appear. Then, re-enter Directory Opus, and save the configuration (described later).

To re-enter Directory Opus, simple click the left mouse button in the

iconified window, then press the right mouse button. If you wish to quit Directory Opus without going back into it, just click the close gadget at the far left of the iconified window.

Note that if you have the configuration option Iconify type (described later) set to hidden, the only way to re-enter Directory Opus is with the hotkeys. These default to CTRL-SHIFT-ALT (the CTRL key, the left SHIFT key and the left ALT key help simultaneously), although this may be changed in the configuration (explained later).

install Disk/Install menu

This function allows you to make a disk bootable. All bootable disks have special information written on the first sector of the disk. Without this, the disk will not boot.

The DF0: gadget can be clicked on to cycle through drives 0 to 3.

If the Check gadget is selected, the disk will not be installed. Instead, the already existing bootblock will be compared with the standard one. If the bootblock is non-standard, you may want to run a virus checker program to check for the possible presence of a virus.

Like **Format** and **Diskcopy**, Install allows you to select multiple disks to Install or check. If you are checking the bootblocks of multiple disks, the result are displayed in a condensed form in the status bar. If, however, you are checking only one bootblock, the result is displayed in full.

lastsaved

This function causes Directory Opus to read in the last saved configuration file. If you have modified the configuration, and have not yet saved it, you are asked for confirmation before this is done.

lplay PLAY gadget (rmb)

Unlike the play function, the lplay function plays sound files continuously, requiring a mouse button click to proceed to the next one.

playst

Similar to the playcommand, playst attempts to play sounds only as SoundTracker/NoiseTracker modules. It allows you to play these modules even the filename does not start with "mod".

makedir MAKEDIR gadget

This function allows you to create a new sub-directory in the active directory window.

move MOVE gadget

This allows you to move all selected files and directories from the active directory window to the inactive window. The file or directory will be copied to the destination, and the original will be deleted.

newcli

This function opens a CLI from within Directory Opus, using the output window defined in the configuration. The new CLI will be separate from Directory Opus, so to end the CLI you must type endcli, as usual.

This function is also called by the run and execute functions, if they are invoked and there are no files selected.

nncopy COPY gadget (rmb)

This function copies all selected files and directories in the active directory window to the inactive directory window, and allows you to specify a new name for each file/directory.

nnmove MOVE gadget (rmb)

This function moves all selected files and directories from the active directory window to the inactive window, and allows you to specify a new name for each file/directory. The file or directory will be copied to the destination under the new name, and the original will be deleted.

none NONE gadget

This function unselects all files and directories in the active directory window.

parent PARENT gadget

This will read the parent directory of the directory open in the active directory window.

There are also two hidden parent gadgets; one for each directory window. They are hidden in the outermost borders (on the left of the screen and the right of the screen) of the directory windows.

pauseprint

This command will pause any print operation that may be in process, and unpause a paused print operation.

play PLAY gadget

This will attempt to play each selected file. If the file is an 8SVX format IFF sampled sound (of the type generated by AudioMaster, for instance), the speed and other information from within the file will be used. This will play any file, though, and if the file is not 8SVX, it will assume a speed of 10000 samples/second.

This function also plays SoundTracker modules, provided the streplay.library file is present in the LIBS: directory. Modules are recognised by the mod. prefix in the filename; if the filename does not have this prefix, it will not be played.

Each file is played once only, as opposed to the Iplay function, which plays each file continuously.

print PRINT gadget

The print routine is a full-featured text formatter. There are several configuration items in the print requester :

Page length: This is the length of each page, in lines.
Left margin: This is the position of the left margin, in characters.
Right margin: This is the position of the right margin.
Tab: This is the number of spaces a tab character is equivalent to.

You can have a header and/or a footer printed at the top and bottom of each page if you wish. Both the header and footer can consist of the title, the current date and the page number. You may select any combination of Title, Date and Page # you wish. If none are selected, the header or footer is not printed.

You may also select the typestyle of the header and the footer, by clicking on the gadgets that initially say Normal. These allow you to cycle through the various typestyles available : Normal, Bold, Italics, Underline, Doublestrike and Shadow. Note that some printers may not have all these typestyles available.

You may enter a title (to be used in the header and footer) in the Title string gadget. Leaving this blank will use the name of the file as a title.

The *Output* gadget allows you to specify where you want the output of the print process to go to. This defaults to PRT:, but you may choose to have the output sent to a disk file. In this case, simply enter the path and the name of the file you wish to print to.

The gadget marked *Normal* at the bottom of the print requester operates in the same way as the header and footer typestyle gadgets, except that it determines the typestyle for all text other than headers and footers.

The gadget marked *Draft* allows you to specify whether all text is printed in draft mode or NLQ mode, headers and footers are printed in draft mode with the rest of the text in NLQ, or headers and footers in NLQ with the rest of the text in draft.

The gadget next to this one determines whether or not the last page of the file to be printed is followed by a formfeed character, ejecting it.

To start the print operation, select the *Print* gadget.

This function will print all selected files, one at a time. If you select only one file to print, the print routine will be started up as a separate process, allowing you to continue working with Directory Opus. To cancel this type of print, simply select the print function again. A requester will appear

asking if you want to continue with the print or halt it. This requester will also appear if you attempt to quit Directory Opus while a print operation is in place, as you cannot quit until the print has finished.

To cancel a print that has not been spooled, simply press the left and right mouse buttons in the normal way.

Note that even if you abort a print, the printer may not actually stop for sometime. This is because most printers have buffers, some quite large ones, which store data faster than it can be printed.

printdir Disk/Print dir menu

This function allows you to print the directory in the active directory window. You are able to choose what information you want to print (file sizes, protection bits, dates and comments). This function is not started as a separate process, unlike the print command. Printdir also allows you to specify the output file to which the directory is printed. This can be PRT: (i.e., your printer, the default) or a disk file.

protect PROTECT gadget

This function allows you to modify the protection bits of the selected files and directories in the active directory window. If any directories are selected, you are asked whether you wish the files within them to be protected as well.

For each entry, you are presented with a requester displaying the protection bits currently set for that entry. The protection bits are H (hidden), S (script), P (pure), A (archived), R (readable), W (writable), E (executable) and D (deleteable).

To toggle bits on or off, click on the appropriate gadgets, or press the appropriate keys.

To set the protection bits of all selected entries you should select the All gadget from the protection requester; choosing Okay will set the protection bits a file at a time.

quit Project/Quit menu

This function will exit Directory Opus (providing there is not a print operation happening at the moment). If the configuration has been modified and not saved, you are asked if you wish to save it first.

read READ gadget

This function allows you to read selected files. At the bottom of the screen the name of the file is displayed, along with the current position within the file in lines and as a percentage, and also the total number of lines in the file.

The mouse is used to move through the file. Press the left mouse button to turn scrolling on or off. The direction and speed of the scroll are governed by the position of the mouse pointer in the window. The mouse pointer will become invisible while the text is scrolling.

No scrolling occurs if you have the mouse in the center of the window. To scroll forward, move the mouse down until the text starts to scroll (providing there is actually more than one page of text). The further down you move the mouse, the faster the text will scroll. This procedure is reversed for backwards scrolling.

The gadgets in the bottom-right of the screen also allow you to move around the file. Most gadgets will repeat if held down for any length of time.

The up and down arrows move up and down a line at a time. The U and D gadgets move up and down a page at a time. The T and B gadgets move to the top and bottom of the file.

You can also use the cursor keys to move up and down. As before, in conjunction with SHIFT these will move a page at a time, and in conjunction with CTRL will move to the top and bottom.

To jump to a specified line, click on the number showing the current line position (immediately left of the word Lines). You can also press the J key. To jump to a specified percentage, click on the number immediately left of the % sign.

To search for a string, click on the S gadget or press the S key. You can

use full pattern matching in this search.

To print the current file, click on the P gadget (or press the P key). To print only the current page, press the C key (this has no gadget equivalent).

To leave the text viewer, click on the X gadget, or press either X, Q or ESC.

If more than one file was selected, the next one will be read when you exit. To exit without reading the next file, press the right mouse button.

relabel Disk/Relabel menu

This function allows you to change the name (label) of the disk that is open in the active directory window.

This function is also by default the right mouse button function of the RENAME gadget.

rename RENAME gadget

This allows you to rename all selected files and directories. You may use a * to specify a pattern. If no *'s are used, you are asked for a new name for each individual file.

To add a prefix or a suffix (or both) to all selected entries, use a * in the second string gadget. For example, to add HIRES as a prefix and .pic as a suffix to all selected entries, enter HIRES*.pic.

To change a prefix or a suffix (or both) of all selected entries, use a * in the top string gadget. For example, to change HIRES to MEDRES as a prefix, and .pic to .iff as a suffix, enter HIRES*.pic in the top string gadget, and MEDRES*.iff in the bottom string gadget.

rescan R gadget (rmb)

This causes the directory in the active window to be re-read. This has exactly the same effect as activating the directory string gadget and pressing return, without modifying the directory name. This function is available as the right mouse button function of the R (re-select) gadget. It

can also be added to your gadgets or menus with *rescan.

reselect R gadget

This function will reselect all entries that were selected before the last operation was initiated. The entries are reselected only if they still exist, and even if the buffer containing them is not displayed currently.

root ROOT gadget

This function reads the root directory of the active directory window. There are also two hidden root gadgets; one for each directory window. They are hidden in the outermost borders (on the left of the screen and the right of the screen) of the directory windows. With the left mouse button, these gadgets perform the parent function; they will load the root directory when selected with the right mouse button.

run RUN gadget

This function allows you to run each selected file in turn, providing that file is executable. This will have much the same effect as if you had double-clicked on the file's icon, or run it from the CLI. A requester will appear, asking for arguments (should the program require any).

saveconfig

This function saves the configuration to the file that it was read from initially.

search SEARCH gadget

This function allows you to search all selected files for a specified string, which can optionally contain wildcards. If a match is found, you are told which file contains the string, and given the option to enter the text viewer and read the file. If you opt to do this, a search for the string is automatically initiated once the file has been read by the text viewer.

If any directories are selected, all the files within those directories are also

searched.

select S gadget

This allows you to select files and directories in the active directory window using wildcards. All standard wildcards are supported (*,~,? etc..) as well as the AmigaDOS wildcards #? and ?.

show SHOW gadget

This function is very versatile indeed. It will display IFF ILBM pictures and brushes, Workbench icons and fonts.

If the file is an IFF ILBM, it will be displayed using information from the file. Directory Opus will correctly show most pictures and brushes, including overscan, extra halfbrite (EHB), HAM (4096 color) pictures, and Dynamic HiRes pictures created by MacroPaint (from Lake Forest Logic).

If the picture contains color cycling information, pressing TAB will toggle color cycling on or off. You can press P to print the picture. Should you wish the mouse pointer to be visible, in order to point at some part of the picture for instance, press the . key.

The show routine will also display icons. If the file has a .info suffix it is assumed to be an icon. If the icon has an alternate image, pressing return will toggle between the two.

To display a font, you need to enter the actual font drawer, and show the size file. That is, the file 8, 12, 19, etc., NOT the .font file.

smartread

This function invokes the text viewer in the same way as the read function. However, it checks the file for binary characters, and if any are found, the file is hexread instead.

stopst

This functions stops any SoundTracker module that is playing.

toggle ALL gadget (rmb)

This function reverses the state of all files and directories in the active directory (ie, all selected entries are deselected, and vice versa).

unbyte BYTE gadget (rmb)

If the directory size for a directory is already displayed, then the directory will not be re-scanned even if it is selected and the byte function is invoked. This could mean that the directory size is inaccurate, especially if any files have been created or deleted in it by another program. Selecting one or more directories, and then selecting the unbyte function will cause the directory sizes (if displayed) to disappear. You can then reselect the directories, invoke the byte function, and they will be re-scanned.

version Project/Version menu

This function displays in the status bar the current versions of Kickstart, Workbench and Directory Opus.

Chapter 6
Configuration

Directory Opus uses a separate program to modify the configuration. This results in a savings of approximately 70K of memory, as the configuration program need only be in memory when it is being used. As you will very rarely need to call upon the configuration program once you have your configuration finalized, this can be a great advantage.

If you do have plenty of memory, however, Directory Opus can be configured to load the configuration program on start-up. In this case, it is run from memory each time the configure function is invoked, rather than loaded from disk. This can be much faster, especially if you have a floppy-based system. The configuration program, ConfigOpus, can only be run from within Directory Opus. It can not be run separately, or when Directory Opus is not running.

The configuration items are grouped together under the headings Operation, Gadgets, Menus, Drives, File types, Functions, Screen, System and Store Config.

Operation

These configuration items affect how Directory Opus operates. The options available under this heading are :

Copy

These flags allow you to configure how the copy function operates.

Check destination free space causes Opus to check that the destination directory has enough free space in it before it attempts to copy files/ directories. If not, Opus will alert you via a requester and give you the option to abort. If this flag is selected and you choose to copy directories that do not have sizes displayed for them, Opus will ask if you want a byte count on these directories before seeing whether or not they will fit in the destination.

Keep original datestamp, *Keep original protection bits* and *Keep original comment* indicate whether you want the datestamp, protection bits and comments are retained when files are copied.

Set archive bit will set the A bit on all files copied.

Delete

These flags allow you to configure how the delete function operates.

Ask before commencing delete will ask for confirmation before the delete process begins.

Ask before deleting files and *Ask before deleting non-empty directories* will ask for confirmation before deleting every file, and every non-empty directory, respectively.

Set delete protection bit will set the D bit on all files that do not have it set. This means that the files protected against deletion will be deleted anyway.

Error check

This allows you to determine how Directory Opus handles DOS errors.

Disable DOS requesters will disable AmigaDOS requesters that would appear on the Directory Opus screen (disk is write protected, etc). Instead, you will only see an error message appear in the status bar.

Display error requester causes Directory Opus to open its own error requester when an error is encountered. This requester contains more information than the standard AmigaDOS requester does.

General

Auto diskchange

If you have the root directory of DF0:, DF1:, DF2: or DF3: open in either of the directory windows, and you change the disk, then the new disk will be automatically read. This allows you to scan through different disks without having to manually read them each time.

Display info

Version 1.xx of Directory Opus displayed the protection bits, datestamp and comment of a file in the status bar. Directory Opus now displays these in the directory windows, and keeps a count of selected files, etc. in the status bar instead.

If you prefer the old style display, turn Display info on. All file information will be displayed in the status bar as you select entries, as well as in the directory windows. With this turned on, you can use the byte function to view the count of selected entries.

Expand Paths

When enabled, this feature will cause all paths to be expanded to their full names. For example, if this flag was enabled and you entered s: in the directory gadget, this might be expanded to Workbench1.3:s/. This also works with drive gadgets: the drive name would be replaced by the volume name in the directory string gadget. Note: Users should avoid using disks with identical volume names to avoid confusion.

File doubleclick

Directory Opus has a special feature not found on many other directory utilities. If this option is selected, Directory Opus will interrogate a file that is doubleclicked on (with the left mouse button), and will perform the default action for that file type.

You may define your own filetypes, and specify what action is performed when a file is doubleclicked. However, there are also several "hard-coded" file types. These are IFF ILBM pictures and brushes, icons, fonts, IFF 8SVX sound files, and executables (programs). If a file does not match any of the user-defined filetypes, it is checked to see if it is one of the above files. In that case, the default action for these file types is performed.

If a file does not match any filetypes, user-defined or otherwise, it is treated as a text file, and read. If the file contains any non-text characters, it will be hexread.

Load configuration program

This flag indicates whether you want the configuration program to be loaded automatically when Directory Opus is run, and to remain in memory

all the time, or whether you want it to be loaded each time you select the configure option in Directory Opus.

Smart parent

If this option is enabled then whenever parent or root is chosen, the directory buffers are searched for the parent or root directory, before the directory is actually read again. This can save a lot of needless disk access.

Icons

This allows you to specify how you want Directory Opus to handle icons.

Create icons with directories causes Directory Opus to create a drawer icon whenever you create a directory with the makedir function.

Perform all actions on icons as well will cause whatever happens to a file to happen to that file's icon as well. For instance, if you delete the file DATA, DATA.info will be deleted too (if it exists). If you rename the file FROG to BUFFALO, then FROG.info (if it exists) will be renamed as BUFFALO.info.

Select icons automatically produces similar results to Perform all actions on icons as well, except that whenever you select a file, its associated .info file (if it exists) is also selected.

Replace

These flags allow you to specify how Directory Opus replaces already existing files, during a copy or similar operation.

Always replace files will cause all files already existing to be replaced.

Never replace files will cause all files already existing to be left untouched.

Replace only older files will cause all files already existing to be replaced, only if the new file has a later datestamp than that of the old file.

Ask before replacing will ask you whether you want to replace an already existing file.

Separate

This allows you to determine how files and directories are displayed.

Mix files and directories causes directories and files to be interspersed, and sorted as one.

Display directories first causes directories to be sorted separately from files, and displayed at the top of the list.

Display files first causes directories to be sorted separately from files, and displayed at the end of the list.

Sort files

This allows you to configure how files and directories are sorted.

Sort alphabetically causes files to be sorted by name.

Sort chronologically causes files to be sorted by date.

Sort numerically causes files to be sorted by size.

If none of these are selected, files and directories are not sorted and are displayed in hash order (the order they are read off the disk).

Update

These options allow you to define how the Directory Opus screen is updated during certain procedures.

Update free disk space causes the free space display in the disk name display at the top of the active directory window to be updated whenever a file is copied to, or deleted from the directory. Otherwise, the display is only updated at the end of a complete operation.

Scroll directory window to follow operations causes the directory windows to scroll, ensuring that the current entry being acted upon in an operation is always visible. The display is returned to its original position when the operation is complete.

Redraw more than a quarter of a page can greatly increase the speed of moving through directories. Ordinarily, if you use the sliders to move a directory LESS than a complete page in either direction, the directory window will scroll a file at a time. It is only if you move a complete page or more that the window will be redrawn. Redrawing is considerably faster, especially in hi-res mode, but is not as pleasing aesthetically. This flag will cause Opus to redraw the window if you move more than a quarter of a page in either direction.

Gadgets

This section allows you to configure the gadgets.

There are 42 physical gadgets, located at the bottom of the Directory Opus screen. There are two banks of these gadgets; in a similar manner to the drive gadgets, only one bank is visible at a time. To access the other bank of gadgets, press the right mouse button over the clock at the bottom of the screen.

Each gadget can also have a right mouse button equivalent, which results in a total of 168 gadget functions.

The gadget configuration screen displays the gadget bank at the top of the screen. The *Other bank* gadget at the bottom allows you to toggle between the two gadget banks.

The *Swap* gadget allows you to exchange the positions of any two gadgets, even from one bank to the other.

To modify the configuration of a gadget, select it from the gadgets at the top of the screen. You are presented with various items you can change.

Gadget name

This is the name that appears on the gadget.

Function

This is the function that is called when this gadget is selected with the left mouse button. See the section on call functions later in this manual for more information.

Other function

This is the function that is called when this gadget is selected with the right mouse button. See the section on call functions later in this manual for more information.

The other function has its own flags and settings. Because of this, you can only edit one function at a time (before, both the first and second function strings could be edited at the same time). In the bottom left corner of the gadget editing screen is the *Other func* gadget. This allows you to toggle (in much the same way as the Other bank gadget does) between the first and second function of each custom gadget.

The only information the second function does not use is the Text color and BG color fields.

You are also able to give the second function an optional Gadget name. If you do, then this name is displayed whenever the gadget is selected with the right mouse button.

Stack size

This is the stack size (in bytes) that the function will use. The default and minimum is 4000 bytes. The documentation of the program you are running should advise you of the best stack size to use.

Priority

This allows you to determine the priority at which the function runs. The default is 0. You should not use anything lower than -10, or higher than 10, except for certain programs (the documentation of the program should advise you of appropriate priorities).

Close delay

This allows you to set the number of seconds after the function has finished before the output window (if any) closes. The default is 2, which means that the output window will close two seconds after the function has terminated. If you set the close delay to 0, the output window will close immediately. If you set it to -1, it will wait for you to press the left mouse button before closing.

Text color

This is the pen color that the gadget name is drawn in. It defaults to 1. You can modify this by either typing the pen color in, or by selecting this string gadget and clicking on one of the colored gadgets on either side of the screen.

BG color

This is the pen color that the background of the gadget is drawn in. It defaults to 0. By judiciously changing the colors of your gadgets, it is possible to create some attractive displays. It is also useful, as it allows you have your "destructive" gadgets (delete, etc) in red, and your "non-destructive" gadgets in green, or similar. You can modify this by either typing the pen color in, or by selecting this string gadget and clicking on one of the colored gadgets on either side of the screen.

Shortcut key

This is the key code of the keyboard equivalent for this gadget. Consult the list of key codes in Appendix B.

Qualifier

This is the qualifier code of the keyboard equivalent for this gadget. Consult the list of qualifiers in Appendix B.

Output window

If this is enabled, an output window will be opened every time this gadget is used. The window will open on the Directory Opus screen, unless the *Workbench to front* option or the *Run asynchronously* option is enabled. In this case, it will open on the Workbench screen, and the Workbench screen will be automatically brought to the front.

The window is opened using the handle specified in the Functions section of the configuration.

Output to file

This option redirects all output from the function to a temporary file in the T: directory, which is then read via the text viewer. This allows you to thoroughly read the output of a program, and even print it.

Note that if you are sending output to a file, the function can not receive input from the keyboard.

Workbench to front

This causes the Workbench screen to be brought to the front when the function is selected.

Directory Opus to front

This causes the Directory Opus screen to be brought to the front when the function has terminated. It is used in conjunction with *Workbench to front.*

If Run asynchronously is turned on, this flag will have a slightly different effect. In this case, if Output window is turned on, the output window will be opened on the Workbench screen as is normal with *Run asynchronously*, but the Workbench screen will not be brought to the front.

Run asynchronously

If this is on, the function will be launched as a new process, and Directory Opus will not wait for it to return. If this is the case, and an output window is specified, the output window opens on the Workbench screen.

CD source

If this is turned on, the current directory of the custom function will be set to the current source directory (active directory window).

CD destination

If this is turned on, the current directory of the custom function will be set to the current destination directory (inactive directory window).

Do all files

This causes the function to act on each selected entry in turn, instead of just the first entry. This is used for commands that do not support multiple filenames on the command line, where {F} (see *Recursive directories,* next paragraph) to send all selected entries would not work.

Recursive directories

Normally, whenever a {f} or {F} or similar sequence would result in the name of a directory being included in the call function, Directory Opus would treat the directory the same as a file. In other words, the function would not act recursively on all files within the directory.

If this option is enabled, the names of all files within the directory, and within sub-directories within the directory, and so on, are included in the call function. This allows the function to act on all files in the directory, and not just the directory itself.

Reload each file

This function could be used with an editor, for example. It causes Directory Opus to rescan a file after it has been acted upon by a function, and updates the size, datestamp, comment and protection bits of the file. You can therefore reflect changes in size, for instance, made by a text editor to a file.

Auto iconify

This option causes Directory Opus to enter iconified mode when the function is launched. This could be useful in low-memory situations. It only works with Run asynchronously.

Don't quote filenames

All custom functions, including gadgets, menus, filetypes, and archiving functions have a new flag called *Don't quote filenames.* Whenever Opus sends a filename to a custom function, with {f} and {o} (and {f}, {o}, etc...) flags, it is now enclosed in quotation marks, This allows you to have external programs accepting filenames with spaces in them. However, some old or ill-written programs will not interpret the quotation marks correctly. If you find this is the case with any program, simply select the Don't quote filenames flag.

Filenames will not be enclosed in quotes unless the character immediately preceding them is a space or a tab (or other white space character). So for instance, if you had he following string as a part of a custom function:

... {fu} "{d}{o} ...

The {fu} would cause the first selected entry to be added (but not deselected, due to the u), and the filename would be enclosed in quotes (unless Don't quote filenames was enabled). The second part of the string, {dr}{o}, would send the path name destination directory, followed by the name of the file used in the {FU}. This second usage of the filename would NOT be enclosed in quotes, as it is immediately preceded by the destination path (and not a space or other white space character). Therefore, to ensure the pathname sent by the {d}{o} is enclosed in quotes, you should have:

... {fu} "{d}{o}" ...

This is fairly complicated and if you still do not understand it, don't worry about it. You will never need quotes around filenames unless either the filenames or the paths have spaces in them (and the use of spaces is discouraged). If you never use spaces in filenames, there will never be a problem.

Executable

Clicking on the gadget that says Executable will see it change to read Batch, and again to read ARexx. Executable means that it is a program, and will be run as one. Batch means that it is a batch file, such as you might run with the execute function, or with the execute command from the CLI. ARexx indicates that the function is an ARexx script (see discussion below of the ARexx interface). The control sequences in the call function (see above) are only taken notice of if the function is defined as an Executable program.

Menus

This section allows you to configure the menus. There are 100 menus in Directory Opus.

The menu configuration screen displays the five menu strips, with the menu titles at the top.

The *Sort* gadget at the bottom allows you to alphabetically sort each individual menu. The Swap gadget allows you to exchange the positions of any two menus.

To modify a menu title, click on the title at the top of each menu strip. You

will then be able to edit the title, up to 14 characters long.

To modify the configuration of a menu, select its gadget from the 100 displayed. The screen will clear, and you will be presented with various items you can change.

The configuration of each menu is performed in the same way as the configuration of the gadgets, except that there is no Second function field.

Drives

The drives section allows you to configure the drive gadgets. There are 24 drive gadgets, arranged in four banks of six.

The drives configuration screen displays the 24 drive gadgets at the top. The Drives gadget at the bottom causes the gadgets to be assigned the names of all the devices and assigned directories (up to 24) present in your system. This is also what happens in the default configuration.

The Swap gadget allows you to exchange the positions of any two drive gadgets.

To define a drive gadget, click on the appropriate gadget at the top of the screen. You are presented with several options.

Drive name

This is the name that will appear in the gadget.

Drive Path

This is the actual AmigaDOS path that will be loaded whenever this gadget is selected. It can be a device such as DF0:, RAM:, etc. or a directory such as DH0:Text/Letters.

Text color

This is the pen color the text in the drive gadget is drawn in. You can modify this by either typing the pen color in, or by selecting this string gadget and clicking on one of the colored gadgets above the Drive name gadget.

BG color

This is the pen color that the background of the drive gadget is drawn in. You can modify this by either typing the pen color in, or by selecting this string gadget and clicking on one of the colors above the Drive name gadget.

Short Cut Key

This is the key code of the keyboard equivalent for this drive gadget. Consult the list of key codes in Appendix B.

Qualifier

This is the qualifier code of the keyboard equivalent for this drive gadget. Consult the list of qualifiers in Appendix B.

File types

Directory Opus allows you to define your own file types. Another name for a file type is a file class. Examples of file types are LHArc archives, animation files, and AmigaBasic programs.

File types are a unique and very powerful feature of Directory Opus. Using file types, you can configure Directory Opus to play animations when they are doubleclicked, to load a database when a database file is selected, to load a music program whenever a music file is doubleclicked, and so on. You may define as many file types as you want.

The File types screen displays the list of file types that Directory Opus recognizes. By default, it recognizes LHArc, Arc and Zoo files. It uses the definitions of these file types to list and extract these archives.

If there are more than sixteen file types known, only sixteen are displayed at a time. To access the previous or next sixteen, click on the up and down arrows at the right of the display.

Each file type is displayed along with its file pattern and character recognition string. These are explained below.

The *Swap* gadget allows you to exchange the positions of two file types in the list. This can be very useful; for instance, if you have defined a file

type, and then wish to define a sub-set of this as another file type (eg, an IFF picture file, and a medium-res IFF picture file), the sub-set must precede the original in the list. Otherwise, files will be matched with the original file type and will never have a chance to match with the sub-set.

To add a new file type, select the *Add* gadget. To delete a file type, select the *Delete* gadget and then the file type you wish to delete.

File types are configured in much the same way as the gadgets and menus are. There are a few additional items you must set, however.

Function type

Each file type can have four separate functions defined for it. This allows you to, for instance, set up an archive file so it can be both listed and extracted.

The four function types are *Auto func*, *Auto func2*, *Doubleclick* and *Clickmclick*.

Auto func and *Auto func2* are the functions called when the auto and auto2 functions are called from a gadget or menu. By default, the ARCHIVE gadget calls the auto function with the left mouse button, and the auto2 function with the right mouse button. Although this gadget is called ARCHIVE, it will actually perform the appropriate action for any file type.

For instance, if you have a word processor file set up to be loaded into your word processor, selecting the ARCHIVE gadget would perform this action.

If you have a lot of file types defined, you may wish to create new gadgets with different names to call the auto and auto2 functions. Or, you may wish to change the name of the ARCHIVE gadget to AUTO or similar.

Doubleclick is the function called whenever a file matching this file type is doubleclicked. The default configuration is set up so that LHArc, Arc and Zoo archives are listed automatically when they are doubleclicked.

Clickmclick is the function called whenever a file matching this file type is selected, and then the other destination window is clicked upon, all within the doubleclick time. This is the same as click-m-clicking a sub-directory to make it open in the other directory window.

The default configuration is set up so that LHArc, Arc and Zoo archives are extracted to the destination window automatically when they are click-m-clicked.

File type

This is the name of this file type; the name that is displayed in the main file types window.

File pattern

This is a wildcard pattern allowing you to define which files match this file type. For instance, *.lzh would match all .lzh files, and so would define LHArc archives. mod.* would match all SoundTracker modules.
Under the default configuration, the last file type in the list, Default, has a file pattern of *. This has the effect of matching all files not already matched. Therefore, you can configure this file type to load any files of unknown type into the text viewer when they are doubleclicked. The default configuration does this, and it will also copy any file that is click-m-clicked, to the destination window. This provides a very quick way to copy a single file.

Char recognition

If you are not sure that a filetype will always match a given file pattern, such as an IFF picture, you may define characters within the file to be matched upon.

The character recognition string consists entirely of hexadecimal numbers, and gives offsets and characters to match at those offsets within files.
This string is defined in the following manner :

offset,characters to match,offset,characters to match,etc...

For instance, all executable program files have hex 00, 00, 03 and F3 as the first four characters in the file. Therefore, to define a file type as an executable file you would enter the character recognition string :

0,000003F3

This means look at offset 0, and try to match the characters 00, 00, 03 and F3.

A more complex example; to match a program crunched by PowerPacker, you could enter the string :

30,2054D1C8,40,2F08D1FC

As a program crunched by PowerPacker is an executable, the example above (0,000003F3) would also pick up PowerPacker files. However, this would pick up all other executables as well. In this case, you would need to have the PowerPacker file type listed before the general executable file type.

To specify to match on any character, use a . character. For instance,

1A2,21D2.A010

means go to offset 1A2 hex, and match the characters 21, D2, anything, A0 and 10.

The offset can be any number of hex characters long; characters to match must consist of two hex digits (unless it is a . character to match on anything).

You may have as many offsets defined in the string as will fit. The character recognition string is checked after the file pattern; that is, if a file matches a file pattern but not the character recognition, it will still pass as that file type. It will also pass if it doesn't match the file pattern but does match the character recognition.

Action description

This is the string that will appear in the status bar whenever this action is performed on a file. For instance, Listing LHArc archive appears under the default configuration whenever you doubleclick or use the ARCHIVE gadget to list an LHArc archive.

Call function

This is the function that is called for this action. See the section on call functions later in this manual for more information.

If for some reason you do not want a function called for this action, you may enter a # (hash) symbol for the call function. If you wish a requester to appear for this action, enter the text of the requester after the # mark. For

example, to find out which programs have been crunched by PowerPacker, you might want the call function to read :

#The program % has been crunched by PowerPacker.

This will display a requester containing the above text. The % sign will be replaced by the name of the file.

The rest of the configuration of file types (Stack size, Priority, Close delay and the ten flags) is identical to the configuration of gadgets.

For the four function types (Auto func, Auto func2, Doubleclick and Clickmclick), only File type, File pattern and Char recognition apply to all. The other configuration items must be set separately for each function type.

If a chosen file is not matched by any of the file types, Directory Opus has some hard-coded file types which it will use. If the file is an icon, a font file or an IFF picture, it will be shown. If the file is an 8SVX sound file or a SoundTracker module, it will be played. If the file is an executable program, it will be run. If the file is a system-configuration file, Preferences will be set from that file.

If the file still does not match any of these, it will be matched with any file type that has * as a file pattern. This will, of course, match any file. If no file type exists with * as a file pattern, it will be loaded into the text viewer and displayed. It will be displayed in hex if it contains any binary characters.

The file type Default is setup in the default configuration with a file pattern of *. Its functions for Auto func and Auto func2 (not that these will probably be used often) are read and hexread. Its function for Doubleclick is smartread - that is, if you double click on a file that doesn't match any other file types, it will be read either as hex or ASCII (depending on the type of characters it contains). Its function for Clickmclick is copy; you can therefore copy a single file by simply click-m-clicking it to the opposite directory.

Functions

The Functions section allows you to define the items that are used for launching programs via the run command. It also allows you to define the setup of the archiving functions.

Default run

This section deals with the launching of programs.

Output command

The Output command is the AmigaDOS command used to launch programs that have output windows. It defaults to NewCLI. If you wish to use a shell, you could change it to NewShell. If you wish to use a WShell, you could change it to NewWsh.

Output window

This is the handle that all output windows are opened under. This is also used to open a CLI by the newcli command, and by the run command when no files are selected.

The output handle allows you to specify the size, position and title of the output window. It follows the standard AmigaDOS format, namely :

DEVICE:LeftEdge/TopEdge/Width/Height/Title

DEVICE	... this is the device to use. It is normally CON, but under AmigaDOS 1.3, it may be NEWCON (which will open a Shell instead of a CLI).
LeftEdge	... the left edge of the window (x coordinate, in pixels).
TopEdge	... the top edge of the window (y coordinate, in pixels).
Width	... the width of the window in pixels.
Height	... the height of the window in pixels.
Title	... the title of the window.

The *Stack size*, *Priority* and *Close delay* parameters are all identical to those used by the gadgets, menus and file types. The flags, too, are identical, however the *Do all files*, *Recursive directories* and *Reload each file* options are not present (as these are illogical in this case).

Archive

While the default file types allow you to list and extract files from archives, they do not provide a way to create and add files to archives. These settings do. They are called by the archive function, which by default is attached to the ARCHIVE A gadget.

Directory Opus allows up to six archive types to be configured. The default configuration sets it up for LHArc, Arc and Zoo - you may edit these and add three more of your own.

Archive type

This gadget allows you to cycle through the six archive types and modify them. However, the position this gadget is left in when you exit the archive configuration also indicates which archiver you wish to use with the archive function.

Archiver name

This is the name of the archiver, and is what appears in the Archive type gadget.

Call function

This is the function to be called to create an archive of this type. Please refer to the section on call functions later in this manual for more information.

The Stack size, Priority and Close delay parameters and the twelve flags are all identical to those used by the gadgets, menus and file types.

Screen

This section allows you change the appearance of the Directory Opus screen.

Palette

This allows you to set the colors of the Directory Opus screen. The boxes numbered from 0 upwards allow you to select which pen color to modify.

Adjust the color by sliding the proportional gadgets to adjust the red, green and blue components.

Colors

This allows you to modify the pen colors of almost all of the components of the Directory Opus screen. To modify a pen color, you can either enter the appropriate number, or select the string gadget you wish to modify and then click on one of the colored boxes.

3-D Boxes

This allows you to change the pen colors of the top and left edges of all "three dimensional" boxes, and the bottom and right edges.

Status text

This allows you to change the pen colors of the text and background of the status bar at the top of the screen.

Files

This allows you to modify the colors used to display files in the directory windows. Files text color and Files background color are the colors used for the text and background of unselected files. Selected text color and Selected background color are the colors used for the text and background of selected files.

Directories

This allows you to modify the colors used to display directories in the directory windows. Dirs text color and Dirs background color are the colours used for the text and background of unselected directories. Selected text color and Selected background color are the colors used for the text and background of selected directories.

Clock

This allows you to set the pen colors of the screen clock at the bottom of the Directory Opus screen.

Requesters

This allows you to modify the background color and text color used in all requesters (other than the about requester).

Disk names

This allows you change the pen colors used to display the disk names and free space at the top of both directory windows. Disk name color and Disk name background are the colors used for the inactive directory window. Selected color and Selected background color are the colors used for the active directory window.

Sliders

This allows you to change the pen color used to display all proportional gadgets (sliders).

Arrows

This allows you to set the pen colors used for the background color and foreground color of the arrow gadgets on the Directory Opus screen.

Tiny gadgets

The tiny gadgets are the B, R, S and A gadgets in the center of the Directory Opus screen, and the ?, E, C, I and Q gadgets at the bottom-right of the screen. This allows you to modify the background and foreground colors of these gadgets.

Depth

This allows you to modify the number of colors the Directory Opus screen uses. The default is four colors; you may select eight or sixteen colors as well. Selecting a "deeper" screen (one with more colors) uses more chip memory and results in a slightly slower display time (scrolling of directories becomes slower, etc).

Also available is the option *Workbench screen*. This allows you, amazingly enough, to open Opus in a window on the Workbench screen. You can use the width and height parameters of the screen size gadget to specify the size of the window. When open on the Workbench screen,

Opus ignores any changes to the palette that you have made and also ignores the Interlace/Half-height/Productivity flags. To determine where on the screen the Workbench window of Opus should be positioned, simply position it and save the configuration (this is similar to the way you determine the position of an iconified window.

Screen size

This allows you to specify the size of the Directory Opus screen.

Workbench size will make the Directory Opus screen the same size as the Workbench screen.

Interlace will turn interlace mode on. This doubles the effective height of the screen, but can result in a disturbing flicker. The correct color combinations, polarized sunglasses or an Amiga 3000 or FlickerFixer can fix this problem.

Half-height also turns interlace mode on, but positions the Directory Opus screen halfway down the monitor. This allows you to have a full sized screen (thanks to interlace mode) but with the screen behind still (partially) visible.

If you are using Kickstart 2.0, there are two additional flags:

Super-Hires will enable Super-Hires mode (double the horizontal width).
Productivity will enable Productivity mode (provided your monitor can cope with it). This doubles the vertical resolution, but without the flickering effect of interlace. Both these modes have a limit of four colors, so any colors you have defined above pen 3 will be ignored.

The *Screen width* and *Screen height* gadgets allow you to modify the actual size of the screen. If you turn one of the flags on, say for instance, Interlace, you would then need to modify the Screen height value to take advantage of the increased height of the screen. That is, if your Screen height was defined as 200 lines, and you turned Interlace on, you would then change the Screen height to 400 lines.

A *Half-height screen* is just that; half the height of a normal interlaced screen. Therefore, if you went from a standard screen to a Half-height screen, you would not need to modify the Screen height.

Font

This option allows you to specify the fixed-width, 8 point font you wish to use on the Directory Opus screen. Proportional fonts do not work, and neither do any fonts that are not 8 points in size.

Arrows

This allows you to change the position of the arrow gadgets in relation to the slider gadgets.

There are three possible choices for each pair of arrow gadgets :

```
========<>  Arrows at the end (or bottom) of the slider
<>========     Arrows at the start (or top) of the slider
<=========>    One arrow at the start and one at the end
```

The default for all three pairs is the first choice. The arrows you modify are the directory window up and down arrows, the directory window left and right arrows and the buffer left and right arrows.

System

These options are general configuration items.

Auto ARexx

This allows you to specify the name of an ARexx script that you wish to be automatically run when Directory Opus starts. It must include the complete pathname. The ARexx interface is fully described in a later section.

Directories

This allows you to specify three things: The pathnames of directories that are to be loaded automatically into the left and right directory windows when Directory Opus starts, and the number of directories (buffers) that are kept in memory.

The number of buffers may be any number from 2 to 100. It describes the

total number of buffers desired; each directory window will have half that number of buffers.

Help file

This allows you to specify the file that Directory Opus reads its help messages from. The default is s:DirectoryOpus.HLP. It must include the complete pathname. If this is left blank, or the file can not be found, the help option will not function.

Hotkeys

This allows you to specify the key code and qualifier of the hotkeys. The hotkeys are a combination of keys that you press, and will do one of three things. If Directory Opus is iconified, it will uniconify Directory Opus. Otherwise, it will bring the Directory Opus screen to the front and to the top. If the screen is already at the top and at the front (and active), it will put it to the back.

The default hotkeys are CTRL-SHIFT-ALT, but you may change this to any combination you like.
To specify no code (as is the default) and to just have a combination of modifying keys, enter FF as the code.

Refer to Appendix B for a list of key codes and qualifiers.

Iconify type

This allows you to choose what the iconified window is displayed as. The choices are a *24 hour clock* (military time), a *12 hour clock*, *No clock* (a small window with just a title in it), or No window (completely non-existent).

If the window is displayed as a clock (either 12 or 24 hour) the date and amount of free memory will also be displayed.

Also, if *CPU monitor* is enabled, Directory Opus will display the current CPU usage (as a percentage) in the clock at the bottom of the screen, and in the clock in the iconified window (if enabled). You may discover some programs (especially communication software) are slowed down slightly by the CPU monitor, but it does not usually cause a problem.

Icons

If you do not like the default icons that are generated for tools, projects and directories using the addicon function, you may specify the name of your own icons to use. These must include the complete pathname.

Default tool allows you to specify the default tool of the project icon created by the addicon function. This is only effective if you are using the build-in tool icon. For example, you may set the default tool to c:Fullview. Then, whenever you double-click (from Workbench) on an icon you have added to a tool, Fullview will be invoked.

Priority

This setting allows you to change the priority at which Directory Opus runs. The default priority is 0, which means that it peacefully coexists with everything else. Setting the priority higher means that Directory Opus will not be slowed down much by other tasks, but the performance of other tasks will suffer. It is not suggested that you set the priority higher than 10.

Screen clock

This allows you to choose whether the screen clock (the clock display at the very bottom of the Directory Opus screen) is displayed in 24 hour or 12 hour format.

Show

This allows you to configure how the show routine functions.

Black background between pictures

If this is enabled, you will see a blank screen after each picture has been displayed, while the next picture is loading. Otherwise, the Directory Opus screen will be displayed in between each picture.

Show delay

This setting allows you to show pictures as a simple slideshow. If set to 0, pictures will wait for you to press the left mousebutton before continuing. If Show delay is set to something other than 0, the pictures will continue to

display (one after the other) after the specified time in seconds.

Fade delay

This setting allows you to configure the speed of the fading in and out of pictures. If it is set to 0, there will be no fading at all. The default is 2 (the value, by the way, indicates in 50ths of a second the time between each change in color). The maximum is 10.

Show Free

This option allows you to choose how the free space that is displayed alongside the disk name is measured. The choices are *Bytes free*, where the number of bytes free is displayed, *Kilo/Megabytes free*, where the free space is displayed in either kilobytes or megabytes (depending upon the size), *Blocks free* (the number of blocks free, as will also be shown by the AmigaDOS info command), or *Percentage of space free*, where the free space is displayed as a percentage of the total size.

Show pattern

This allows you to prevent certain files and directories from being displayed in the directory windows.

Hide files with hidden bit set tells Directory Opus to take notice of the H bit; any files that have this protection bit set will not be displayed.

Show pattern is a wildcard pattern that all files are checked against. If the file does not match this pattern, it is not displayed.

Hide pattern is a wildcard pattern that all files are checked against. If the file matches this pattern, it is not displayed.

Store config

The following options are available under the configuration storage menu:

Default loads the initial configuration that comes with Directory Opus

Last saved reloads the most recently saved configuration file

Load brings up a file requester giving you the opportunity to load any previously saved config file

Save as lets you save an Opus configuration as a reloadable file

Quick Save

This option on the main config screen simply saves the configuration to the current configuration file.

Chapter 7:
Launching External Programs

All gadgets, menus, file types and archive definitions have one thing in common: they all allow you to run external programs. That means that as well as defining a gadget to activate, for instance, the built-in diskcopy routine, you can define a gadget to load and run an external disk copier as well.

To launch an external program, you specify a call function. Gadgets have two of these - they are called Function and Second function, but they work in exactly the same way.

Call functions can either call internal routines, or run external programs. To specify an internal routine, simply enter the name of the function (listed in chapter 5) preceded by a * character (asterisk).

For example, to make a gadget run the internal disk copying routine, you would enter *diskcopy as the call function.

However, if you wish to call an external program, the call function contains the AmigaDOS name (including complete path) of the function that is to be called. This can contain certain control sequences, allowing you to do special things with the function.

{aTitle:Default} ... The function will ask you for arguments whenever this sequence appears. The Title string is the title that will appear on the argument requester. You can just omit the Title string, in which case a default title will be used. The Default string (preceded by a colon) is the default string that is to appear in the argument requester. This too can be omitted. If the default string begins with the sequence [o] or [f], the name of the last used file will be inserted as the default string. [o] will insert just the name, and [f] will include the complete path name. This is only effective is there has been a {o} or {f} sequence prior to the {a}. For example, it would be possible to have your own rename gadget with the following call function:

```
rename {f} {ar} {aEnter new filename: [o] }
```

The [o] or [f] may be followed by other text, but not preceded by it.

{d} ... This will insert the path name of the destination directory window into the call function.

{dr} ... The same as {d} except that it causes the destination directory to be re-read at the end of the procedure.

{Dr} ... This will cause the destination directory to be re-read, but the pathname will NOT be inserted into the call function.

{f} ... The name of the first selected entry will be inserted into the call function for this particular time. The entry is deselected.

{fu} ... The same as {f} except that the entry is not deselected.

{F} ... The names of ALL selected entries are inserted into the call function at this point. AmigaDOS limits the length of command lines to 256 characters. Therefore, if the length of a call function exceeds 256 characters because of a {F} sequence (or for any reason), a new line in the script file (see below) is added, containing the next lot of filenames. This means that you can select as many entries as you wish with a {F} sequence, and they will all be accepted.

{Fu} ... The same as {F} except that the entries are not deselected.

{o} ... The same as {f} except that only the name of the first selected entry is used, without the pathname.

{ou} ... The same as {o} except that the entry is not deselected after it is used.

{O} ... The same as {F} except that only the names of the selected entries are used, without the pathname.

{Ou} ... The same as {O} except that the entries are not deselected after they are used.

{s} ... This will insert the path name of the source directory window into the call function.

{sr} ... The same as {s} except that it causes the source directory to

be re-read at the end of the procedure.

{Sr} ... This will cause the source directory to be re-read, but the pathname will NOT be inserted into the call function.

For example, the default call function to add files to an LHArc archive looks like this :

```
c:LHArc -x a {dr}{aEnter archive name} {O}
```

The {dr} sequence tells Directory Opus to insert the destination path. It is immediately followed by a request for user input; the requester that appears will have the title Enter archive name.

The {O} sequence tells Directory Opus to add the names of all selected files and directories. Certain flags for the default LHArc archive function are also set (see chapter 6 for information on these flags); one of these is CD source. Since the source directory will become the current directory, the {F} sequence is not needed. The {O} sequence includes file names without paths, but as these files are in (what is to become) the current directory, this does not matter.

So imagine you have selected two files to add to a new archive (you could, of course, have selected any number you like). These files are called Picture1.iff and ReadMe.txt. The destination directory is open; imagine its path is RAM:temp/.

If you were to select the archive function under these circumstances, you would be asked to enter the archive name. Imagine you enter test as the name of the archive. Directory Opus would then build and execute the following command :

```
c:LHArc -x a RAM:temp/test Picture1.iff ReadMe.txt
```

Once the archive has been created, the destination directory will be re-read (the r in the {dr} sequence indicates this). This will update the display to reflect the new file that has been created.

For more examples of call functions, examine the default call functions for LHArc, Arc and Zoo in the File types section as well as in the Archive section of Functions, in the configuration.

The way custom functions are launched has changed radically from prior versions of Directory Opus. Before, custom functions were run as programs; that is, the program you entered on the call function line was simply run. This led to some severe limitations, namely :

a) Maximum function length of 256 characters.
b) No input could be typed at the output window.
c) The current directory of the function was set to the boot device.
d) The stack of the function was always 4000 bytes.
e) The priority of the function was always 0.
f) Functions run asynchronously could not have output (or input).

In the new approach to custom functions, temporary script files are created in the T: directory. This has many advantages over the previous method, in that the above limitations no longer exist. One of the few disadvantages is a slightly longer operation time, as the script file must be created each time the function is used. This extra time is negligible if the T: directory is assigned to a hard drive, or the RAM disk. If you do not have a hard drive, you should include the line Assign T: RAM: in your startup-sequence. If the T: directory is not assigned, Directory Opus will use RAM: instead.

Another disadvantage is that an extra program is needed. This program is called DOpusRT, and can be found in the C directory of the distribution disk. This is only 3K long, and should be copied to your C: directory.

The DOpusRT program is called at the end of the script file, and communicates with Directory Opus to inform it when the script file has finished running. This is needed for functions that are not run asynchronously. It also doubles as a new wait command, which is needed for the adjustable close delay.

To take advantage of the current directory changing, adjustable stack and priority, you will need the commands cd, stack and changetaskpri in your C: directory. You will also need the commands run, execute, failat, newcli and endcli. These are almost always present in your C: directory anyway, so you should not need to install them.

There are a couple of other points to be aware of. Because custom functions are run as a script file, they can be aborted like any other script file by pressing CTRL-D. This is only possible if you have an output window selected.

If the script file is aborted, the output window effectively turns into a CLI,

and as such, you must type endcli to remove the window.

Another point is that if the function was not run asynchronously, Directory Opus will be frozen, waiting for the signal back from the DOpusRT program. If you abort the script file before it reaches DOpusRT, Directory Opus will never return. In this frozen state, you will see Opus' *busy pointer*, which looks like a wristwatch with a moving second hand. You can force Directory Opus to unfreeze by activating the Directory Opus window and pressing SHIFT-ESC (the shift key in conjunction with the escape key). This also allows you to turn any non-asynchronous function into an asynchronous function, while the function is running.

Note that when using NewCLI or NewShell as the Output command, an error such as unknown command will abort the script file as well. You should therefore make sure that all your custom functions do not try to run nonexistent programs.

Chapter 8:
On-line help

Directory Opus provides comprehensive "on-line" help. You can assign help text to any gadget or menu. Directory Opus also comes with a help file, which contains help for the default gadgets and menus.

The help file is a text file which contains a number of messages. If the help mode is enabled, and a gadget or menu which has a help message associated with it is chosen, the help message is displayed in a requester. Also displayed is the keyboard equivalent (defined by the Shortcut key and Qualifier configuration items), so if you ever forget what the shortcut key for a gadget or menu is, you have only to access its help text.

Each help message in the help file is preceded by a function identifier. This tells Directory Opus which function the following help message refers to. The function identifier is a string, which consists of either the name of the gadget or menu, or the function string of the gadget or menu.

Each function identifier is preceded by a * character (asterisk). Each help message ends with a ^ character (carat).
For example, the default help text for the about function looks like this :

```
**about

This function displays some information about the program,
including the current version number, copyright information,
and how to contact technical support.^
```

The **about string is the function identifier. It starts with a * character. It is then followed by the string *about, which is the default function string of the Project/About menu. The help message then begins on a new line.

Each help message can have an unlimited number of lines in it, but the length of each line should not go over 78 characters (or, the width of the Directory Opus screen in characters less 2). The help message then ends with a ^ character, not on a new line.

The help message does not need to include a string for the keyboard

shortcut; if the gadget or menu has a keyboard shortcut defined, Directory Opus will insert this information automatically when you access the help message.

Therefore, if you wish to define help for an internal function, the function identifier consists of two * characters followed by the name of the function.

If you wish to define help for an external function, however, you have two choices.

The function identifier can consist of either the function string (as above), or the name of the gadget or menu. You would also use the name of the gadget or menu as the identifier if the function string was the same as for other gadgets (for example, you might have two gadgets with different names, both calling the auto function).
For instance, the help message for the ARCHIVE gadget, which calls the auto function, starts like this :

> *archive
>
> This function allows you to automatically list or extract archives created by the programs Arc, Zoo and LHArc. The type of archive is recognized automatically.

As you can see, the identifier only has one * character on it. The archive string following it refers now to the name of the gadget.

The help file is also used to store the text for the error help function. To define your own text for an error code, the function identifier consists of a * character, a ! character, and the error code itself. For instance, *!103 refers to error code 103 (no free store).

Examine the default help file if you are still unsure of how it is laid out.

Chapter 9:
General Information

If you click-m-click with the left mouse button, first on the diskname of the active directory window, and then on the diskname of the inactive directory window, the active directory window will be copied to the inactive one.

If you click-m-click with the right mouse button in the same way, the two directory windows will be swapped.

Directory Opus will automatically decrunch text crunched with PowerPacker, in the text viewer and for the print function. For this to have effect, you must have the powerpacker.library in your LIBS: directory. This is not supplied with Directory Opus, but can be obtained from most public domain libraries.

It is highly recommended that you use the FF program on the Workbench 1.3 disk to speed up text output. This can greatly increase the speed of Directory Opus.

Directory Opus has another flag that you can use when you run it. The -g flag will prevent Directory Opus from complaining if there is already a copy of Directory Opus running. The equivalent ToolType from Workbench is CHECK=0. This can allow you to have your startup-sequence or similar run more than one copy of Directory Opus. In conjunction with the -c flag (CONFIGFILE ToolType), you can have several different copies of Directory Opus, all with different configurations, running at once.

Incidentally, the CHECK and ICONSTART ToolTypes require boolean arguments. For instance, CHECK=1 would enable the checking described above. This is the default, however, so in this case the ToolType could just be omitted.

Chapter 10:
ARexx Interface

Rexx is a programming language, the Amiga version of which is called ARexx. ARexx, by William S. Hawes, takes the idea of a programming language one step further, by supporting the fast and easy passing of messages between an ARexx program (often called a script) and an external program with an ARexx interface. This allows any ARexx program to control an external application, and effectively makes the external application programmable.

Directory Opus sports a complete ARexx interface. Almost every function accessible to the user can be controlled from ARexx, including all the configuration items.

There are no examples of ARexx scripts given here. There are several examples on the Directory Opus distribution disk, in the AREXX directory.

All Directory Opus ARexx commands are listed below. A few notes beforehand, though :

-- The ARexx port is called dopus_rexx.

-- Arguments are surrounded by { } if required, or [ ] if optional.

-- Words surrounded by < > are optional keywords.

-- All commands return results, if any, in the ARexx RESULT variable.

-- Any commands that would normally put up a requester (i.e., select) will do so if no arguments are specified.

-- For any commands that take a filename as an argument, the filename must be the name of an entry that is present in the active directory window, and should not include a pathname.

-- All ARexx commands are shown below in lowercase. They are, however, not case sensitive.

Many of these commands are identical to the functions listed in chapter 5. Some do, however, take arguments, or return a result; those that do are

described in detail here.

abortprint

This aborts the current print process and returns 1. If no file is currently being printed, it returns 0.

about

addbuffers [number]

This command will add the specified number of cache buffers to the device open in the active directory window. If number is not specified, the user will be asked for the amount.

addicon [filename]

This command will add an icon to all selected entries, or just filename if it is specified.

all
archive

arexx [command]

This command allows you to execute the supplied command. If command is not supplied, the user is prompted for one.

assign [devicename]

This allows you to assign a logical device name to the active directory. If devicename is not specified, the user is asked for one.

auto
auto2
bufferlist
byte

cd [directory]

This allows you to set the current directory. If directory is not specified, the user will be asked for the pathname.

checkabort

This command allows you to monitor, from an ARexx script, whether the user has pressed the left and right mouse buttons together (to abort). It also resets the short flag every time it is called. If you plan to use the checkabort routine, you should call checkabort at the start of the script (to reset the abort flag), before actually using it to check the abort status.

clearbuffers

clone [filename]

This command will clone all selected entries (or just filename if it is specified) in the active directory window. The user is asked for each new filename in turn.

comment [comment [<recurse>]]

This will add a comment to all selected entries. If one is not specified, the user is asked for one. If the recurse keyword is included, directories will be commented recursively.

configure
contst

copy [filename]

This copies all selected entries to the destination directory, or just filename if it is specified.

date [date time]

This command allows you to set the system time and date. If a date and time are not specified, the user will be prompted for them. If they are specified (and both must be if any), the format is DD-MMM-YY HH:MM:SS.

datestamp [<recurse>]

This command will set the datestamp of all selected entries to the current date and time. If the recurse keyword is included, directories will be datestamped recursively.

defaults

delete [filename]

This command will delete all selected entries in the active directory, or just filename if it is specified.

dirtree [window]

This command will map out a directory tree in the active directory window, or the specified window (either 0 or 1).

diskchange [window]

This creates a diskchange event for the device open in the active window (or the specified window).

diskcopy [source destination(s) [<verify>]]

This command will diskcopy from source to destination, with verify turned on if the verify keyword is specified. If no source or destination are specified, the diskcopy requester is opened. Arguments may include multiple destinations. For example, *diskcopy DF0: DF1: DF2:* would result in the disk in DF0: copying to DF1: and DF2:.

diskinfo

encrypt [password]

This command will encrypt all selected files with the specified password. If a password is not specified, the user is asked for one.

errorhelp [code]

This will open the error help requester and display help for the specified error code. If no code is specified, the user is asked for the error number.

execute

format [drive name [<verify> <quick> <noicons>]]

This command will open the format requester, unless a drive is specified. If a drive is specified, the format will begin immediately. The verify, quick and

noicons options may also be specified, as may be the name. Format supports multiple destination drives and volume names. Example:

format DF0: Blankdisk DF1: DF2: workdisk verify noicons

would format DF0:, giving it the name "Blankdisk", format df1:, giving it the default name, and format df2: giving it the name "workdisk", with the formatting verified and no icons created on any of the disks.

getdevices
help

hexread [filename]

This will hexread all selected files, or just filename if it is specified.

hunt [filename]

This command will hunt through all selected directories for the specified filename. If one is not specified, the user is asked for one.

iconify

install [drive [<check>]]

If no drive is specified, the install requester is opened. Otherwise, the specified disk is installed. Multiple disks may be specified to be installed. If the check keyword is specified, the disk's existing bootblock will be checked. In this case, RESULT will contain 0 if the bootblock is non-standard, 1 if it is standard, 2 if it is not bootable, or -1 if an error occured.

lastsaved

lplay [filename]

This will play all selected files continuously, or just filename if it is specified. Each file will require a mouse button click to proceed to the next one.

makedir [dirname]

This will create a directory in the active directory window, using dirname if it is supplied.

move [filename]

This moves all selected entries to the destination directory, or just filename if it is specified.

newcli [handle]

This function will open a new CLI window. If no handle is specified, the Output window in the configuration will be used.

nncopy [filename [newname]]

This allows you to copy all selected files, and will prompt you for a new name for each. If filename is given, just that file is copied. If newname is given with filename, newname is used as the new name.

nnmove [filename [newname]]

This allows you to move all selected files to the destination directory, and will prompt you for a new name for each. If filename is given, just that file is moved. If newname is given with filename, newname is used as the new name.

none
parent

pauseprint

This command will pause or unpause any print operation that may be in process. It returns 1 if the print is now paused, 2 if the print has been unpaused, or 0 if there is no print operation in process.

play [filename]

This command plays all selected files, or just filename if it is specified.

playst

print [filename]

This command will print all selected files, or just filename if it is specified.

printdir [destination [<size> <protect> <date> <comment>]]

This command will print the active directory. If any keywords are included, only those items are printed. The optional destination may be either prt: (default) or a disk file, in which case you would specify the path/filename.

protect [protectionflags [<recurse>]]

This command will set the protection bits of all selected files. Unless protection is supplied (in the form HSPARWED, or bits thereof), the user will be prompted for the protection bits. If the recurse keyword is included, any selected directories will be protected recursively.

quit

read [filename]

This will read all selected files, or just filename if it is specified.

relabel [name]

This will relabel the disk open in the active directory window, using the supplied name (if one is supplied). If a name is not supplied, the user will be asked for one.

rename [oldname newname]

This command will rename all selected entries in the active directory window. If oldname and newname are not specified, the user is asked for each new name. You may rename only one filename by entering that filename as oldname. Oldname and newname may include wildcards.

reselect

rescan [window]

This causes the active window's directory to be re-read. Optionally you can specify the window to be re-read. A zero (0) means the left window while an one (1) means the right window.

root
run
saveconfig

search [searchstring]

This will search all selected files in the active directory window for a specified string. If one is not included, the user will be asked for one.

select [pattern [<onlyfiles| onlydirs>]]

This command will select entries based upon the supplied pattern. If either the onlyfiles or the onlydirs keyword is specified, only files or only directories are selected.

show [filename]

This command attempts to show all selected files, or just filename if it is specified.

smartread [filename]

This command will read all selected files, or just filename if it is specified. If any of the files contain binary characters, they will be hexread instead.

stopst
toggle
unbyte
version

The following commands are not listed as functions in chapter 5; they are solely ARexx commands.

addfile [name size type seconds comment protection #ofblocks display]

This command adds an entry to the active directory window. Name is the filename, size is the file size in bytes, and type is the file type (less than 0 for a file, greater than 0 for a directory). seconds is the datestamp of the file in seconds from midnight on January 1st, 1978. comment is an optional file comment, and protection is the protection bits of the file, in the form HSPARWED. #ofblocks is the file size in blocks; this is not used yet, and so can really be set to 0 (at least for the moment). display is a boolean indicating whether you want the window display updated or not. This command returns a boolean indicating success or failure.

busy <on| off>

This command turns the busy pointer either on or off, depending upon the supplied keyword.

checkabort

This command returns a boolean indicating whether the user has pressed the left and right mouse buttons (to abort). It also resets the abort flag. You should call this routine to initially clear the abort flag, before actually using it to check the abort status.

copywindow [win1 win2]

This command will copy the contents of window win1 to window win2.

displaydir [window]

This command will update the display for the active directory window, or the specified window (either 0 or 1).

dopustoback

This command sends the Directory Opus screen to the back.

dopustofront

This command brings the Directory Opus screen to the front.

drive {drivegadget}

This command will act as if you had just clicked on the specified drive gadget (from 0 to 23). It reads the directory assigned to that drive gadget, and returns the directory name in the RESULT variable.

fileinfo {name} [separator]

This command returns information on the entry called name (which must be in the active directory window). The information returned is separated by separator (default separator is space), and consists of the filename, the size in bytes, the size in blocks (not used yet), the file type (less than 0 indicates a file, greater than 0 indicates a directory), the selected status (boolean), the date (days), the date (seconds), the comment and the

protection bits.

The date is returned as a number of days since January 1st, 1978, and as the number of seconds since midnight. The protection bits are returned in the format HSPARWED (with a - character where the bit is not set).

gadget {gadget}

This command will activate the specified gadget, where gadget is a number from 0 to 191. Gadgets 0 to 47 are the gadgets on the first bank of gadgets; gadgets 48 to 95 are those on the second bank of gadgets. Gadgets 96 to 143 are the right mouse button functions of the first bank of gadgets; gadgets 144 to 191 are the right mouse button functions of the second bank.

If the number you specify is a multiple of eight (including zero), it is ignored; these are then the drive gadgets, and you choose them with the drive command.

getall [separator]

This command returns, in the RESULT variable, the names of all entries in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getdirs [separator]

This command returns, in the RESULT variable, the names of all directories in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getentry {num}

Returns the name of entry number num.

getfiles [separator]

This command returns, in the RESULT variable, the names of all files in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getnextselected

This command returns the name of the first selected entry (in the RESULT variable) in the active directory window. The entry is NOT deselected afterwards (so if you don't deselect it yourself, you will only ever get returned that name).

getselectedall [separator]

This command returns, in the RESULT variable, the names of all selected entries in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getselecteddirs [separator]

This command returns, in the RESULT variable, the names of all selected directories in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getselectedfiles [separator]

This command returns, in the RESULT variable, the names of all selected files in the active directory window. If separator is specified, the names are separated by separator. The default separator is a space. You should be prepared to handle a long string.

getstring [message]

This command will prompt the user to enter a string (maximum 256 characters), displaying message if it is specified.

menu {menu} {item}

This allows you to activate your menus. menu is the menu number, from 0 to 4, and item is the item number from 0 to 19.

nextdrives

This command simply flips to the next 6 drive gadgets.

otherwindow

This is the same as pressing the space bar, and activates the inactive directory window. The number of the new active window (0 or 1) is returned in the RESULT variable.

patternmatch {pattern} {string}

This will return a boolean value to indicate whether the supplied string matches the supplied pattern (which may include full wildcards).

redraw

This command will completely redraw the Directory Opus screen. The screen is closed, and then re-opened. This provides a way to implement changes made to the configuration via status 31 (see below).

removefile {name} [display]

This command removes the specified entry from the active directory window, and returns a boolean indicating whether the entry was found and removed. display is a boolean value indicating whether you want the display updated or not.

request [message]

This command returns 1 or 0 to a yes/no type request, displaying message if it specified.

scrollh {window} [chars]

This will scroll the selected window (either 0 or 1) horizontally by the specified number of characters (either positive or negative). The new horizontal position is returned in the RESULT variable.

scrollv {window} [lines]

This will scroll the selected window (either 0 or 1) vertically by the specified number of lines (either positive or negative). The new vertical position is returned in the RESULT variable.

selectfile {filename} {selectstate} {display}

This command will select the entry specified by filename, as if it had just been clicked on. If selectstate equals 1, the entry is selected, and if selectstate equals 0, the entry is deselected. The window display is updated if display equals 1, and not if display equals 0.

This command returns the old select state (either 0 or 1) in the RESULT variable.

setprint1 {length} {lmargin} {rmargin} {tabs} {style} {formfeed}

This allows you to configure the print routine.

length	= length of the page in lines
lmargin	= character position of the left margin
rmargin	= character position of the right margin
tabs	= number of spaces a tab is equivalent to
style	= the style the text will be printed in; Normal=1, NLQ=2, Bold=3, Italics=4, UnderLine=5, DoubStrk=6, Shadow=7
formfeed	= 0=eject page, 1=no eject

setprint2 {header} {footer} {hstyle} {fstyle} {title} {output}

This allows you to configure the print routine.

header	= flags to turn on in header (add the chosen ones); 0=no header, 1=title, 2=date, 4=page
footer	= flags to turn on in footer
hstyle	= the style the header will be printed in; as style above
fstyle	= the style the footer will be printed in
title	= the title that will be printed, "" for name of file

swapwindow

This command swaps the contents of the two directory windows.

toptext {string}

This command will display string in the status bar.

The following commands are called status commands. They are accessed with the command status, and may be used to set or return information about the current state of Directory Opus.

status {num} [subnum] [<set> [value]]

This command returns the value for the specifed status command (listed below). If the set keyword is specified, this command will set the status to the specified value (if appropriate for that status command). Most status values can be set.

Some status commands require a sub-number. An example of this is status 4, which returns the total number of files in a specified directory window. The directory window may be either 0 (for the left-hand window) or 1 (for the right-hand window). The command would therefore be status 4 0 for the left-hand window, and status 4 1 for the right-hand window.

Status values are as follows :

1 Current directory

This returns or sets the pathname of the current directory.

2 Version

This returns the current version number (as a string).

3 Active window

This returns or sets the active directory window (either 0 or 1).

4 Number of files {window}

This function returns the total number of files in the specified directory window (0 or 1).

5 Number of dirs {window}

This function returns the total number of directories in the specified directory window (0 or 1).

6 Number of entries {window}

This function returns the total number of entries in the specified directory window (0 or 1).

7 Number of selected files {window}

This function returns the total number of selected files in the specified directory window (0 or 1).

8 Number of selected dirs {window}

This function returns the total number of selected directories in the specified directory window (0 or 1).

9 Number of selected entries {window}

This function returns the total number of selected entries in the specified directory window (0 or 1).

10 Total bytes {window}

This function returns the total number of bytes (sum of the lengths of all files) in the specified directory window (0 or 1).

11 Total selected bytes {window}

This function returns the total number of selected bytes (sum of the lengths of all selected files) in the specified directory window (0 or 1).

12 Select pattern

This returns or sets the current Select pattern (used in the Select command).

13 Directory name {window}

This function returns the name of the directory that is open in the specified window. Setting this to another pathname will read that directory.

14 Disk name {window}

This function returns the name of the disk that is open in the specified window. This cannot be set (use the relabel command to change the disk name).

15 Disk free bytes {window}

This function returns the bytes free on the disk open in the specified window.

16 Disk total bytes {window}

This function returns the total number of bytes (free+used) on the disk open in the specified window.

17 Buffered directory name {buffer}

This function returns the name of the directory that is stored in the specified buffer (from 0 up to 99).

18 Buffered diskname {buffer}

This function returns the name of the disk that is open in the specified buffer (from 0 up to 99).

19 Buffered disk free bytes {buffer}

This returns the bytes free on the disk that is open in the specified buffer.

20 Buffered disk total bytes {buffer}

This returns the total number of bytes on the disk that is open in the specified buffer.

21 Buffer displayed {window}

This returns the number of the buffer that is displayed in the specified window, a number from 0 to 99. Setting this has the effect of buffering back and forth (see earlier sections).

22 Display offset {window}

This returns as a number of lines the offset of the display in the specified window. This can be set, and has the effect of moving through the directory display.

23 Buffered display offset {buffer}

This returns or sets the offset of the display in the specified buffer, in lines.

24 Entries per page

This returns the number of entries that can be displayed per page.

25 Configuration changed

This returns a boolean value to indicate whether the configuration has been changed since it was last saved.

26 Okay string

This returns the name of the Okay string in requesters. This also allows you to select the text of the positive gadget for the request and getstring commands.

27 Cancel string

This returns the name of the Cancel string in requesters. This also allows you to select the text of the negative gadget for the request and getstring commands.

28 Currently iconified

This returns a boolean value indicating whether Directory Opus is currently iconfied.

29 File currently printing

This function returns 0 if there is no print operation in process, 1 if there is, and 2 if there is currently a paused print operation.

30 Top text justification

This returns a value indicating whether the text in the status bar is centered (0), left-justified (1) or right-justified (2). This can also be set.

31 Configuration structure

This returns the decimal address of the configuration structure in memory. The configuration structure is listed in file Structures.H; this then allows you to examine or modify the configuration.

Note: If you modify certain configuration items such as the custom gadget names, the screen size, etc., the screen will not be updated to reflect the change until it is redrawn. To redraw the screen, use the redraw command.

The only configuration items you can not change successfully from ARexx are the number of buffered directories, the name of the help file, and the flag that indicates whether the configuration program is to remain in memory or be loaded each time the configuration function is invoked. These items can only be changed successfully from the configuration program.

Changing the number of buffered directories from ARexx will probably cause a crash. Changing the other two items will not cause a crash, but their new contents will not be taken notice of until the next time the configure function is used.

32 Function 32 *no longer supported*

33 First file type

The filetypes are stored as a linked list of structures in memory; refer to the file Structures.H for the actual structure itself. This status command returns the address (in decimal) of the first file type. To remove a filetype, you should point its parent on to its descendant, and then FreeMem() it. To add one, you should AllocMem() it and link it into the list. You should Forbid() and Permit() while you are doing this.

The four call functions for each file type are also AllocMem()'d separately, in the same way as the call functions for gadgets and menus (refer to status 32 above).

Appendix A:
The Conversion Program

The program Convert on the distribution disk allows you to convert your old Directory Opus v2.xx configuration files to the format understood by the current version. The Directory Opus configuration file is stored in binary format, and so can load and save up to eight times faster than the configuration files of Directory Opus v2.xx.

The Convert program will transfer most of your v2.xx configuration to the current Opus format. Obviously, there are some new features in Directory Opus that did not exist in v2.xx - in these cases, the new configuration will reflect the default state.

There are also some configuration items of v2.xx that can not be brought over to the current format. Unfortunately, any menus in the fourth configurable menu of v2.xx can not be brought to the current format. If you configured the fourth menu of Directory Opus v2.xx, try to swap as many items into the other three menus before converting the configuration. Those items that are lost you will have to reconfigure.

The other items that can not be brought over are the call functions of the archivers. This should not be a problem, as not many people will have changed the default archiving call functions.

To convert your configuration, run the Convert program. The ARP file requester will appear, asking you to enter the pathname of your old v2.xx configuration file. Once you have selected this, you should enter the pathname of the new file you want Convert to create.

You should keep a backup copy of the old v2.xx configuration until you are sure the converted configuration works correctly.

Appendix B:
Key codes and Qualifiers

To specify which key or key combinations you must press for either the hotkeys, or the keyboard equivalents for a gadget or menu, you must specify a key code and a qualifier.

The key code relates to which "real" key you must press. A real key is one that actually does something itself, and doesn't just modify the action of other keys. Examples of "real" keys are the function keys, the letter and number keys, the cursor keys, etc.

The qualifier relates to which modifying keys you press. Modifying keys are those like CTRL, SHIFT, ALT, etc.

Both the key code and qualifier are entered as a hexadecimal value. A table of key codes is below.

To specify no code; that is, just a combination of modifying keys, enter FF as the code (this is the default for the hotkeys).

To specify a combination of modifying keys, you must add each modifying key's code. For example, LEFT SHIFT + CONTROL + LEFT ALT (which is the default qualifier for the hotkeys) equates to 01 + 08 + 10 = 19.

Qualifiers

LEFT SHIFT	- 01	LEFT AMIGA	- 40
RIGHT SHIFT	- 02	RIGHT AMIGA	- 80
CONTROL	- 08		
LEFT ALT	- 10		
RIGHT ALT	- 20		

Key codes

F1 - 50	A - 20	X - 32
F2 - 51	B - 35	Y - 15
F3 - 52	C - 33	Z - 31
F4 - 53	D - 22	
F5 - 54	E - 12	[- 1A
F6 - 55	F - 23	] - 1B
F7 - 56	G - 24	; - 29
F8 - 57	H - 25	' - 2A
F9 - 58	I - 17	, - 38
~ - 00	J - 26	. - 39
1 - 01	K - 27	/ - 3A
2 - 02	L - 28	
3 - 03	M - 37	ESC - 45
4 - 04	N - 36	TAB - 42
5 - 05	O - 18	BACKSPACE - 41
6 - 06	P - 19	DELETE - 46
7 - 07	Q - 10	HELP - 5F
8 - 08	R - 13	RETURN - 44
9 - 09	S - 21	SPACE - 40
0 - 0A	T - 14	CURSOR UP - 4C
- - 0B	U - 16	CURSOR DN - 4D
= - 0C	V - 34	CURSOR LT - 4F
\\ - 0D	W - 11	CURSOR RT - 4E

(- 5A	6 - 2F	
) - 5B	7 - 3D	
/ - 5C	8 - 3E	
* - 5D	9 - 3F	
1 - 1D	0 - 0F	These keys are on the keypad.
2 - 1E	. - 3C	
3 - 1F	- - 4A	
4 - 2D	+ - 5E	
5 - 2E	ENTER - 43	

Default keyboard equivalents

These are the keyboard equivalents that come set up in Directory Opus. These can not be changed.

SPACE	- toggle active directory window
UP/DOWN CURSOR	- scroll up/down
UP/DOWN CURSOR + SHIFT	- move a page up/down
UP/DOWN CURSOR + CTRL	- move to top/bottom
LEFT/RIGHT CURSOR	- scroll left/right
LT/RT CURSOR + SHIFT	- move a page left/right
LT/RT CURSOR + CTRL	- move to far left/right
LT/RT CURSOR + ALT	- buffer left/right
RETURN	- active directory gadget
any character	- scroll to first entry starting with that character
any character + SHIFT	- scroll to first file starting with that character

These are the default keyboard equivalents for the menus. These can all be changed.

Project Menu

HELP	.....	Help!
AMIGA-C	.....	Configure
AMIGA-I	.....	Iconify
AMIGA-Q	.....	Quit

Disk Menu

AMIGA-D	.....	Diskcopy
AMIGA-F	.....	Format
AMIGA-N	.....	Install

These are the default keyboard equivalents for the gadgets. These can all be changed.

CTRL a	.....	ALL
CTRL n	.....	NONE
CTRL p	.....	PARENT
CTRL o	.....	ROOT
CTRL b	.....	BYTE
CTRL m	.....	MAKEDIR
CTRL D	.....	DELETE
CTRL I	.....	CLONE
CTRL c	.....	COPY
CTRL R	.....	RENAME
CTRL M	.....	MOVE
CTRL E	.....	EDIT
CTRL u	.....	RUN
CTRL e	.....	ENCRYPT
CTRL y	.....	PLAY
CTRL s	.....	SHOW
CTRL r	.....	READ
CTRL h	.....	HEXREAD
CTRL i	.....	ADD ICON
CTRL g	.....	ASSIGN
CTRL P	.....	PRINT
CTRL L	.....	ARCHIVE
CTRL v	.....	ARCHIVE A
CTRL C	.....	COMMENT
CTRL d	.....	DATESTAMP
CTRL t	.....	PROTECT
CTRL H	.....	HUNT
CTRL S	.....	SEARCH

In most requesters, you can press the first letter of a gadget instead of clicking on the gadget itself. However, Okay and Cancel have priority over other gadgets, and in this case, the second letter is generally used.

Main Index:

* 7-1, 8-1
^ 8-1
? Gadget 3-14

-c CLI option 1-6
-g CLI option 9-1
-i CLI option 1-5
-o CLI option 1-6

[f] 7-1
[o] 7-1
{aTitle:Default} 7-1
{dr} 7-2
{Dr} 7-2
{d} 7-2
{Fu} 7-2
{fu} 7-2
{F} 7-2
{f} 7-2
{Ou} 7-2
{ou} 7-2
{O} 7-2
{o} 7-2
{sr} 7-2
{Sr} 7-3
{s} 7-2

2.0 Screen Modes 6-22
3D Box Colors 6-20
8SVX Sound 3-5

A Gadget 3-14
Aborting 2-1
Aborting Launched Programs 7-5
AbortPrint Function 5-1
About Function 5-1
Action Description Field 6-16
Active Directory 2-1
Active Window 1-6
Add FileType Gadget 6-14
Add Icon Gadget 3-8
AddBuffers Function 5-1
AddIcon Configuration 6-25
AddIcon Function 5-2
All Function 5-2
All Gadget 2-5, 3-1
Alt-Left/Right Keys 2-3
Arc 6-15, 6-19
Archive A Gadget 3-10
Archive Add 6-19
Archive Bit 1-2
Archive Configuration 6-19
Archive Function 5-2
Archive Gadget 3-10
Archive Type Gadget 6-19
Archiver Name Field 6-19
ARexx Examples 10-1
ARexx Function 5-2
ARexx Gadget 3-14
ARexx Interface 10-1
ARexx Lexicon 10-1
Arrow Colors 6-21
Arrow Configuration 6-23
Assign Function 5-2
Assign Gadget 3-8
Asterisk Character 8-1
Auto ARexx Field 6-23
Auto Directory Fields 6-23
Auto DiskChange Flag 6-2
Auto Func 6-14
Auto Func2 6-14
Auto Function 5-3, 6-14
Auto Function2 5-3, 6-14
Auto Iconify Flag 6-10

B Gadget 3-13
Background Color Field 6-8, 6-13
BG Color Field 6-8, 6-13
Buffer Configuration 6-23
Buffer Gadget 3-13
BufferList Function 5-4
Buffers 1-6, 2-3
Busy Pointer 7-5

Byte Function 5-4
Byte Gadget 3-2

C Gadget 3-14
C: files 7-4
Call Function Field 6-16, 6-19
Carat Character 8-1
CD Destination Flag 6-9
CD Function 5-4
CD Source Flag 6-9
Char Recognition Field 6-15
Character Recognition Field 6-15
Check ToolType 9-1
Clear Buffers Gadget 3-13
ClearBuffers Function 5-4
CLI option -c 1-6
CLI option -g 9-1
CLI option -i 1-5
CLI option -o 1-6
Click-M-Click 1-7, 2-3, 6-14, 9-1
Clock Colors 6-20
Clock Flags 6-24
Clone Function 5-5
Clone Gadget 3-3
Close Delay Field 6-7
Color Configuration 6-20
Color Cycling 3-6
Comment 1-2
Comment Function 5-5
Comment Gadget 3-11
ConfigFile ToolType 9-1
ConfigOpus 6-1
ConfigPath ToolType 9-1
Configuration 6-1
Configure Function 5-5
Configure Gadget 3-14
Control Left/Right Keys 2-4
Control Up/Down Keys 2-4
Control-D Key 7-4
Control-Left Shift-Left Alt 4-3
ContST Function 5-5
Conversion A-1
Convert A-1
Copy Configuration 6-1
Copy Function 5-6
Copy Gadget 3-3
CPU Monitor Flag 6-24
CTRL-D Key 7-4
CygnusED Professional 1-5

Date Function 5-6
DateStamp 1-2
DateStamp Function 5-6
DateStamp Gadget 3-11
Decrypting 3-5
Default FileTypes 6-17
Default Function 5-6
Default GadgeKeys B-4
Default Gadget 6-26
Default Gadgets 3-1
Default Keys B-3
Default Menus 4-1
Default Run Configuration 6-18
Default Tool Field 6-25
Delete Bit 1-2
Delete Configuration 6-2
Delete FileType Gadget 6-14
Delete Function 5-6
Delete Gadget 3-3
Depth Configuration 6-21
Device List Gadget 3-14
Directories 1-1
Directories Configuration 6-23
Directory Buffers 1-6
Directory Entry 1-7
Directory Entry Colors 6-20
Directory Gadgets 1-7
Directory Icon Field 6-25
Directory Opus to front Flag 6-9
Directory Tree 1-3
Directory Tree Gadget 3-14
Directory Window 1-7
DirTree Function 5-7
DirUtil 1-1
Disk Name 1-7
Disk Name Colors 6-21
Disk/Add Buffers Menu 4-5
Disk/Disk Copy Menu 4-4

Disk/Disk Info Menu 4-5
Disk/Format Menu 4-4
Disk/Install Menu 4-5
Disk/Print Dir Menu 4-5
Disk/Relabel Menu 4-5
DiskChange Function 5-7
DiskCopy Function 5-7
DiskInfo Function 5-7
Display Info Flag 6-3
Do All Files Flag 6-9
Don't Quote Filenames Flag 6-10
DOpusRT 7-4
dopus_rexx 10-1
DOS 1-1
DOS Error Help 8-2
DOS Error Help Gadget 3-14
Double-Click 1-7, 2-3, 6-14
Drawer Icon Field 6-25
Drive Configuration 6-12
Drive Gadgets 1-7, 2-2
Drive Name Colors 6-21
Drive Name Field 6-12
Drive Path Field 6-12

E Gadget 3-14
Edit Gadget 3-4
Encrypt Function 5-8
Encrypt Gadget 3-5
Entering a Sub-Directory 2-3
Entry 1-7
Error Check Configuration 6-2
Error Help Gadget 3-14
ErrorHelp Function 5-8
Executable Gadget 6-11
Execute Bit 1-2
Execute Function 5-8
Expand Paths Flag 6-3
External Programs 7-1

Fade Delay Field 6-26
File Double-Click Flag 6-3
File Entry 1-7
File Entry Colors 6-20
File Size 3-2
File Types 1-7
FileName Pattern Field 6-15
FilePattern Field 6-15
Files 1-1
FileType Configuration 6-13
FileType Field 6-15
FileType Gadget 6-11
FileType Priority 6-13
Find File 3-12
Font Configuration 6-23
Font Viewer 3-6
Format Function 5-8
Frozen (Opus status) 7-5
Function Identifier 8-1
Function Type Gadget 6-14
Functions 5-1
Functions Configuration 6-17

Gadget 1-7
Gadget Function Field 6-6
Gadget Name Field 6-6
Gadgets Configuration 6-6
General Configuration 6-2
General Information 9-1
GetDevices Function 5-9

HalfHeight Flag 6-22
Help 8-1
Help File Field 6-24
Help Function 5-9
Help Gadget 3-14
Help TextFile 8-1
Hex Read Gadget 3-8
HexRead Function 5-10
Hidden Bit 1-2
Hidden Parent Gadget 3-1
Hidden Root Gadget 3-1
Hide Hidden Files Flag 6-26
Hide Pattern Field 6-26
Highlight 1-8
Hot Key 4-3
HotKey Codes B-1
Hotkeys Configuration 6-24
Hunt Function 5-10
Hunt Gadget 3-12

I Gadget 3-15
Icon Configuration 6-25
Icon Flags 6-4
Iconify Configuration 6-24
Iconify Function 5-10
Iconify Gadget 3-15
IconStart ToolType 9-1
Inactive 1-8
Inactive Window 1-8
Install Function 5-11
Installing 1-4
InstallOpus 1-4
Interlace Flag 6-22
Introduction 1-1

KeyCodes B-2

LastSaved Function 5-11
LastSaved Gadget 6-26
Launching Examples 7-3
Launching Limitations 7-4
Launching Procedure 7-4
Launching Programs 7-1
Left Mouse Button 2-1
Left/Right Keys 2-4
Left/Right Mouse Button 2-1
LHArc 6-15, 6-19
Load Config Program Flag 6-3
Load Gadget 6-26
LPlay Function 5-11

MakeDir Function 5-12
MakeDir Gadget 3-3
Menu 1-8
Menu Configuration 6-11
Menu Item Configuration 6-12
Menu Strips 6-11
Menu Title 6-11
Menus 4-1, 6-11
Move Function 5-12
Move Gadget 3-4

NewCLI 3-5, 7-5
NewCLI Function 5-12
NewShell 7-5
NNCopy Function 5-12
NNMove Function 5-12
None Function 5-13
None Gadget 2-5, 3-1

On-Line Help 8-1
Open 1-8
Open Directory 1-8
Operation Configuration 6-1
Opus 2.xx A-1
Other Bank Gadget 6-6
Other Func Gadget 6-7
Other Function Gadget 6-7
Output Command Field 6-18
Output to file Flag 6-8
Output Window Field 6-18
Output window Flag 6-8

Palette Configuration 6-19
Parent Function 5-13
Parent Gadget 3-1
Pattern Matching 2-5, 3-4, 3-14
PausePrint Function 5-13
Play Function 5-13
Play Gadget 3-5
PlayST Function 5-12
PowerPacker Files 6-16
Print Function 5-13
Print Gadget 3-9
PrintDir Function 5-15
Priority Configuration 6-25
Priority Field 6-7, 6-25
Productivity Flag 6-22
Project Icon Field 6-25
Project/About Menu 4-3
Project/Configure Menu 4-2
Project/Current Dir Menu 4-1
Project/Error Help Menu 4-2
Project/Help! Menu 4-2
Project/Iconify Menu 4-3

Project/Quit Menu 4-4
Project/Set Date Menu 4-1
Project/Version Menu 4-3
Proportional Gadget 1-7
Protect Function 5-15
Protect Gadget 3-11
Protection Bits 1-2, 3-12
Pure Bit 1-2

Q Gadget 3-15
Qualifier Field 6-8, 6-13
QualifierCodes B-1
Quick Save Gadget 6-26
Quit Function 5-16
Quit Gadget 3-15

R Gadget 2-6, 3-13
Read Bit 1-2
Read Function 5-16
Read Gadget 3-6
Reading a Directory 2-2
Reading a Sub-Directory 2-3
Recursive Directories Flag 6-10
Relabel Function 5-17
Reload each file 6-10
Rename Function 5-17
Rename Gadget 3-4
Replace Flags 6-4
Requester 1-8
Requester Colors 6-21
Rescan Function 5-17
Rescan Gadget 3-13
Reselect Function 5-18
Reselect Gadget 2-6, 3-13
RESULT 10-1
Return Key 2-2
Right Mouse Button 1-8, 2-1, 2-5
Right Mouse Button Gadget 1-8
RMBGadget 1-8
Root Function 5-18
Root Gadget 3-1
Run Asynchronously Flag 6-9
Run Function 5-18
Run Gadget 3-5
Running Opus 1-5

S Gadget 2-5, 3-14
S:DirectoryOpus.HLP 8-1
Save As... Gadget 6-26
SaveConfig Function 5-18
Screen Clock Configuration 6-25
Screen Configuration 6-19
Screen Depth Configuration 6-21
Screen Height Field 6-22
Screen Mode Flags 6-22
Screen Size Configuration 6-22
Screen Width Field 6-22
Script Bit 1-2
Search Function 5-18
Search Gadget 3-12
Searching 2-4
Select Function 5-19
Select Gadget 2-5, 3-14
Selected 1-8
Selecting Entries 2-5
Separate Flags 6-5
Shift Left/Right Keys 2-4
Shift Up/Down Keys 2-4
Shift-Escape Key 7-5
ShortCut key Field 6-8, 6-13
Show Configuration 6-25
Show Delay Field 6-25
Show Font 3-6
Show Free Configuration 6-26
Show Function 5-19
Show Gadget 3-6
Show Pattern Configuration 6-26
Show Pattern Field 6-26
Slider Colors 6-21
Smart Parent 3-1
Smart Parent Flag 6-3
SmartRead Function 5-19
Sort File Flags 6-5

Sort Gadget 6-11
Sound Tracker 3-6
Space Bar 2-1
Stack Size Field 6-7
Startup-Sequence 1-5
Status Bar 1-8, 2-6
Status Text Colors 6-20
StopST Function 5-19
Store Config Gadget 6-26
Store Configuration 6-26
String Gadget 1-7
Sub-Directory 1-3, 2-3
SuperHires Flag 6-22
Swap Drives Gadget 6-12
Swap FileTypes Gadget 6-13
Swap Gadgets Gadget 6-6
System Configuration 6-23

Terms 1-6
Text Color 6-12
Text Color Field 6-8
Text Printer 3-9
Text Reader 3-7
Text Search 3-12
Theory 1-4
Tiny Gadget Colors 6-21
Tiny Gadgets 3-13
Toggle Function 5-20
Tool Icon Field 6-25
ToolType Check 9-1
ToolType ConfigFile 1-6, 9-1
ToolType ConfigPath 1-6
ToolType IconStart 1-6, 9-1

UnByte Function 5-20
UnByte Gadget 3-2
Up/Down Keys 2-4
Update Flags 6-5

Version Function 5-20

Wild-Cards 2-5, 3-4, 3-14
Workbench Depth Flag 6-21
Workbench Size Flag 6-22
Workbench to front Flag 6-9
Write Bit 1-2

Zoo 6-15, 6-19

ARexx Index:

AbortPrint 10-2
About 10-2
AddBuffers 10-2
AddFile 10-8
AddIcon 10-2
All 10-2
Archive 10-2
ARexx 10-2
Assign 10-2
Auto 10-2
Auto2 10-2

BufferList 10-2
Busy 10-9
Byte 10-2

CD 10-2
CheckAbort 10-3
CheckAbort 10-9
ClearBuffer 10-3
Clone 10-3
Comment 10-3
Configure 10-3
ContST 10-3
Copy 10-3
CopyWindow 10-9

Date 10-3
DateStamp 10-3
Defaults 10-4
Delete 10-4
DirTree 10-4
DiskChange 10-4
DiskCopy 10-4
DiskInfo 10-4
DisplayDir 10-9
DOpusToBack 10-9
DOpusToFront 10-9
Drive 10-9

Encrypt 10-4
ErrorHelp 10-4
Execute 10-4

FileInfo 10-9
Format 10-4

Gadget 10-10
GetAll 10-10
GetDevices 10-5
GetDirs 10-10
GetEntry 10-10
GetFiles 10-10
GetNextSelect 10-11
GetSelectedAll 10-11
GetSelectedDirs 10-11
GetSelectedFiles 10-11
GetString 10-11

Help 10-5
HexRead 10-5
Hunt 10-5

Iconify 10-5
Install 10-5

LastSaved 10-5
LPlay 10-5

MakeDir 10-5
Menu 10-11
Move 10-6

NewCLI 10-6
NextDrives 10-11
NNCopy 10-6
NNMove 10-6
None 10-6

OtherWindow 10-12

Parent 10-6
PatternMatch 10-12
PausePrint 10-6
Play 10-6
PlayST 10-6
Print 10-6
PrintDir 10-7
Protect 10-7

Quit 10-7

Read	10-7	Select	10-8
Redraw	10-12	SelectFile	10-13
Relabel	10-7	SetPrint1	10-13
RemoveFile	10-12	SetPrint2	10-13
Rename	10-7	Show	10-8
Request	10-12	SmartRead	10-8
Rescan	10-7	Status	10-14
Reselect	10-7	StopST	10-8
Root	10-7	SwapWindow	10-13
Run	10-7		
		Toggle	10-8
SaveConfig	10-7	TopText	10-13
ScrollH	10-12		
ScrollV	10-12	UnByte	10-8
Search	10-8		
		Version	10-8

Status Table:

(1)	Current Directory	10-14
(2)	Version	10-14
(3)	Active Window	10-14
(4)	Number Of Files	10-14
(5)	Number Of Dirs	10-14
(6)	Number Of Entries	10-15
(7)	Number Of Selected Files	10-15
(8)	Number Of Selected Dirs	10-15
(9)	Number Of Selected Entries	10-15
(10)	Total Bytes	10-15
(11)	Total Selected Bytes	10-15
(12)	Select Pattern	10-15
(13)	Directory Name	10-15
(14)	Disk Name	10-16
(15)	Disk Free Bytes	10-16
(16)	Disk Total Bytes	10-16
(17)	Buffered Directory Name	10-16
(18)	Buffered Disk Name	10-16
(19)	Buffered Disk Free Bytes	10-16
(20)	Buffered Disk Total Bytes	10-16
(21)	Buffer Being Displayed	10-16
(22)	Display Offset	10-17
(23)	Buffered Display Offset	10-17
(24)	Entries Per Page	10-17
(25)	Configuration Changed	10-17
(26)	Okay String	10-17
(27)	Cancel String	10-17
(28)	Currently Iconified	10-17
(29)	File Currently Printing	10-17
(30)	Top Text Justification	10-18
(31)	Configuration Structure Pointer	10-18
(32)	*No longer supported*	
(33)	First FileType Pointer	10-19

INOVAtronics
8499 Greenville Ave. #209B
Dallas, Tx 75231
Phone 214-340-4991
FAX: 214-340-8514

Directory Opus copyright 1990, 1991 Jonathan Potter, published and exclusively distributed by Inovatronics, Inc. CanDo copyright Inovatronics, Inc.